

■ PROJECTS ■ THEORY ■ APPLICATIONS ■ CIRCUITS ■ TECHNOLOGY

NUTS
AND
VOLTS

NUTS AND VOLTS

www.nutsvolts.com
January 2015

EVERYTHING FOR ELECTRONICS

**Build A
Barn Door
TRACKER for Astro
PHOTOGRAPHY**

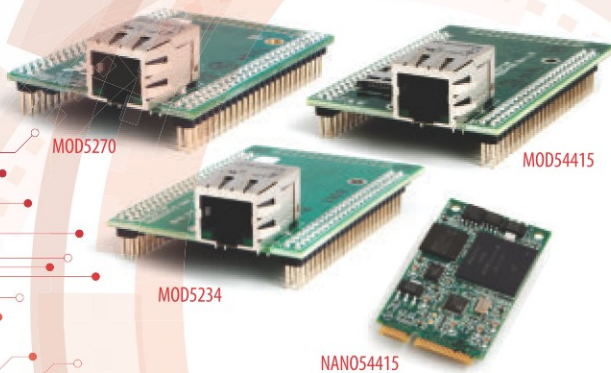
*Roll Your Own
Sounding
Rocket
Flight
Computer*

**Anaren's
New
ATMOSPHERE
Takes Bluetooth
Apps To New Heights**

The Ham's Wireless Workbench

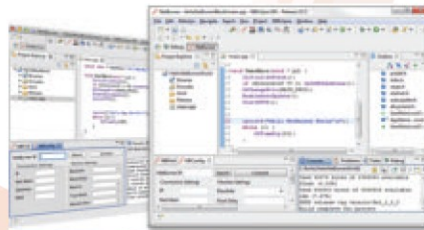
NEW! Whatever your specialty, learning about
radio will help you select, apply, and
use wireless technologies better.

Ethernet Core Modules with High-Performance Connectivity Options



- **MOD5270**
147.5 MHz processor with 512KB Flash & 8MB RAM · 47 GPIO · 3 UARTs · I²C · SPI
- **MOD5234**
147.5 MHz processor with 2MB flash & 8MB RAM · 49 GPIO · 3 UARTs · I²C · SPI · CAN · eTPU (for I/O handling, serial communications, motor/timing/engine control applications)
- **MOD54415**
250 MHz processor with 32MB flash & 64MB RAM · 42 GPIO · 8 UARTs · 5 I²C · 3 SPI · 2 CAN · SSI · 8 ADC · 2 DAC · 8 PWM · 1-Wire® interface
- **NANO54415**
250 MHz processor with 8MB flash & 64MB RAM · 30 GPIO · 8 UARTs · 4 I²C · 3 SPI · 2 CAN · SSI · 6 ADC · 2 DAC · 8 PWM · 1-Wire® interface

Add Ethernet connectivity to an existing product, or use it as your product's core processor



The goal: Control, configure, or monitor a device using Ethernet

The method: Create and deploy applications from your Mac or Windows PC. Get hands-on familiarity with the NetBurner platform by studying, building, and modifying source code examples.

The result: Access device from the Internet or a local area network (LAN)

The NetBurner Ethernet Core Module is a device containing everything needed for design engineers to add network control and to monitor a company's communications assets. For a very low price point, this module solves the problem of network-enabling devices with 10/100 Ethernet, including those requiring digital, analog and serial control.

MOD5270-100IR.....\$69 (qty. 100)	NNDK-MOD5270LC-KIT\$99
MOD5234-100IR.....\$99 (qty. 100)	NNDK-MOD5234LC-KIT\$249
MOD54415-100IR.....\$89 (qty. 100)	NNDK-MOD54415LC-KIT\$129
NANO54415-200IR...\$69 (qty. 100)	NNDK-NANO54415-KIT.....\$99

NetBurner Development Kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications. The kits include all the hardware and software you need to build your embedded application.

➤ **For additional information please visit**
<http://www.netburner.com/kits>

all the inventions waiting
to happen...

CLICKER² STM32

price:
\$39⁰⁰

January 2015

26 A Barn Door Tracker for AstroPhotography

Now, you can be the star and take long exposure photographs of the night sky.

■ By Theron Wierenga



32 Arduino-Based Master Clock

If you've got some old impulse secondary clocks that are lacking a master, build this project and get them working accurately and electronically. Plus, you can repurpose what you learn to control other items using an Arduino and an IGBT MOSFET device.

■ By Joe Fox



40 Flight Computers for Sounding Rockets

Sure, launching model rockets is cool and all, but what if you could incorporate some science and learning into the fun, too. The flight computers described here will help you do that.

■ By Dan Gravatt

46 Soldering Reflow Controller

This product review describes a new kit/system that can help alleviate some of the drudgery involved with building surface-mount printed circuit boards.

■ By Paul Atkinson

Departments

06 DEVELOPING PERSPECTIVES

Just in Time Parts vs. the Junkbox

24 NEW PRODUCTS

25 SHOWCASE

66 ELECTRO-NET

74 NV WEBSTORE

77 CLASSIFIEDS

78 TECH FORUM

81 AD INDEX

Columns

08 Q&A

Reader Questions Answered Here

Answering a question about a previous question, free to air satellite TV, and testing GFCIs properly are discussed this month.

12 PICAXE Primer

Sharpening Your Tools of Creativity

Harnessing the Power of USB.

Construct a couple of simple USB-based power supplies that you can use in your next PICAXE project.

19 The Ham's Wireless Workbench

Practical Technology from the Ham World

NEW!

An Intro and Antennas.

Amateur radio is back in the pages of NV!

Did you know that hams are big players in the Arduino and Raspberry Pi communities?

Even if you're not interested in the full ham radio experience, you might be curious about using non-licensed wireless data links in your projects. Whatever your specialty, learning about radio will help you select, apply, and use wireless technologies better.

53 Smiley's Workshop

Programming • Hardware • Projects

The Arduino Classroom.

Arduino 101/Chapter 12: Data Logger/Part 1. Learn how to store data on an Arduino UNO so you can use battery-operated devices to collect data out in the field.

62 Open Communication

The Latest in Networking and Wireless Technologies

Serial I/O Data Interfaces: Part 1.

Data is transferred serially in most applications.

However, there is definitely more than one way to "send" that cat.

68 The Design Cycle

Advanced Techniques for Design Engineers

Atmosphere Takes Bluetooth Apps to New Heights.

Anaren is bringing Bluetooth "out of the stone age" with their new web-based tool that lets you simultaneously code an embedded Bluetooth monitor/control application and its associated smartphone GUI app.

MIXED SIGNAL OSCILLOSCOPES

4 ANALOG + 16 DIGITAL CHANNELS

RAPIDLY DEBUG COMPLEX MIXED SIGNAL DESIGNS

- USB 3.0
- ULTRA DEEP MEMORY
- SEGMENTED MEMORY
- RAPID TRIGGERS



INCLUDES AUTOMATIC MEASUREMENTS, SPECTRUM ANALYZER, SDK, ADVANCED TRIGGERS, COLOR PERSISTENCE, SERIAL DECODING (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), MASKS, MATH CHANNELS, ALL AS STANDARD, WITH FREE UPDATES . 5 YEAR WARRANTY



	3204D MSO	3205D MSO	3206D MSO	3404D MSO	3405D MSO	3406D MSO
Channels	2 analog, 16 digital			4 analog, 16 digital		
Bandwidth	60 MHz	100 MHz	200 MHz	60 MHz	100 MHz	200 MHz
Buffer memory	128 MS	256 MS	512 MS	128 MS	256 MS	512 MS
Max. sampling rate	1 GS/s					
Signal generator	Function generator + Arbitrary waveform generator					
Digital inputs	100 MHz max. frequency, 500 MS/s max. sampling rate					

Call: 1.800.591.2796 www.picotech.com/pco533



by Bryan Bergeron, Editor

DEVELOPING PERSPECTIVES

Just in Time Parts vs. the Junkbox

Cleaning out my workshop reminded me of when I first started my journey in electronics — tubes were still available at RadioShack. My first ham radio transmitter — a

HeathKit DX-60B — used a 6146B tube final amplifier (power amplifier), in part because it was inexpensive and readily available. Back then, I had a junkbox with a few

dozen tubes, a pound or two of discrete resistors and capacitors, and some miscellaneous hardware. With that, I could repair just about any TV, receiver, or transmitter that I came across or wanted to modify.

Today, things are more complicated, in part because of the vast array of specialized solid-state components and assemblies available. Moreover, the shelf life of these components and assemblies are typically months instead of years or decades.

Don't get me wrong. I look forward to getting my hands on the latest developments in technology. I'm hooked, for example, on the new product announcements featured by SparkFun every Friday, and the broader daily updates on technology from Gizmodo. It's just that it no longer pays to have a sizeable junkbox.

For example, about a year ago, I created a prototype circuit for a government grant application that relied on some high-resolution TFT LCD modules from AdaFruit. However, in the nine months between the grant application and funding, the TFT LCD had been discontinued. As a result, I had to experiment with a new crop of LCDs and newly introduced OLEDs from AdaFruit. I was happy with the new displays, which provided higher resolution, faster response, and no increase in price. However, I had to modify the 3D printer file for the mounting assembly, and rework the software to accommodate the new generation of displays.

Today, after clearing out perhaps 20 lbs of retired ASICs (application-specific integrated circuits), breakout boards, and circuit assemblies, my junkbox consists largely of leaded and SMT resistors, capacitors, and a few 3V and 5V power regulator chips. I've downsized from six shelves of "junk"

ARLCD 3.5-inch Arduino GPU Combo Just \$99.00



Product Specifications: 3.5" color TFT LCD • 320 x 240 Resolution • 65k colors • Touchscreen • Powerful 16-bit microcontroller GPU • 4MB flash memory for storing fonts, bitmaps & macros • USB 2.0 • Overall outline dimensions: 3 x 3 x .9 inches • 6-9V operating voltage • Extremely low power - draws less than 200mA • ARLCD Arduino GUI Library • Arduino UNO R3 Compatible



EarthMake.com

DS1000Z

Digital Oscilloscope

RIGOL
Beyond Measure

Advanced Analysis for Entry Level Applications!



- 50MHz, 70MHz & 100MHz Models
- 1GSa/s Max. Sample Rate
- Serial Triggering & Decode
- Integrated 25MHz Generator
- New DS1054Z Starting at \$399

Visit RigolScope.com
or call 877-4-RIGOL-1

Compare vs. TEK DPO2000 & Save!

NUTS AND VOLTS

EVERYTHING FOR ELECTRONICS

Published Monthly By
T & L Publications, Inc.

430 Princeland Ct.

Corona, CA 92879-1300

(951) 371-8497

FAX (951) 371-3052

Webstore orders only 1-800-783-4624

www.nutsvolts.com

Subscription Orders

Toll Free 1-877-525-2539

Outside US 1-818-487-4545

P.O. Box 15277

North Hollywood, CA 91615

FOUNDER

Jack Lemieux

PUBLISHER

Larry Lemieux

publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ ADVERTISING SALES

Robin Lemieux

robin@nutsvolts.com

EDITOR

Bryan Bergeron

techedit-nutsvolts@yahoo.com

VP OF OPERATIONS

Vern Graner

vern@nutsvolts.com

CONTRIBUTING EDITORS

Joe Pardue

Tim Brown

Fred Eady

Lou Frenzel

Ron Hackett

Ward Silver

Paul Atkinson

Theron Wierenga

Joe Fox

Dan Gravatt

CIRCULATION DEPARTMENT

subscribe@nutsvolts.com

SHOW COORDINATOR

Audrey Lemieux

WEB CONTENT

Michael Kaudze

website@nutsvolts.com

WEBSTORE MARKETING

Brian Kirkpatrick

sales@nutsvolts.com

WEBSTORE MANAGER

Sean Lemieux

ADMINISTRATIVE STAFF

Debbie Stauffacher

Re Gandara

Copyright © 2015 by T & L Publications, Inc.

All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: 430 Princeland Court, Corona, CA 92879.

Printed in the USA on SFI & FSC stock.



to one shelf of discrete components.

As a result, there's less to keep track of, and I know where everything is. Previously, it was a hunt to find that elusive chip or breakout board. Now, my hunting is reserved to online searches.

The two to three day time lag is a significant downside, but no greater than working with components that are no longer supported. So far, this "just in time" parts procurement has worked just fine.

As an aside, I've also shifted from what I once considered standard 4-40

stainless nuts and bolts from suppliers such as BoltDepot, to lighter smaller M2x10 hardware from places such as HobbyKing. I wouldn't think of building a drone or any other compact lightweight device with the old hardware.

However, there's still enough of the heavy duty boards around such as the Arduino UNO microcontroller board to hang on to my supply of 4-40 hardware. I'm sure that eventually — as with my collection of RF vacuum tubes — I'll be tossing these, as well.

NV

ALL ELECTRONICS

www.allelectronics.com

Order Toll Free 1-800-826-5432

DIGITAL PANEL METER

Volts: 3.5-30 Vdc.

Amps: 0-5A. Volts and

Amps on the same compact

digital panel meter. Built in

shunt. Bright, red LED

display. Easy to connect;

three-wire friction-lock connector plugs into back of meter. Snap-mounts into 46 x 27mm (1.80" x 1.05") rectangular cut-out. Black bezel, 47.5 x 28.5mm.

CAT# PM-23

\$17.00
each



30 DETENT ROTARY ENCODER WITH SWITCH

Alps mechanical rotary encoder

with momentary-on push switch.

30 detents. 6mm dia. aluminum shaft.

12mm square body. 7mm dia. x 10mm high

plain bushing. PC pins. CAT# RE-30

\$1.35
each

50 for \$1.15 each



USB BREAKOUT CABLE

Provides two USB

ports as well as

one micro-USB

and one

mini-USB port.

CAT# USB-45

\$5.95
each



12 VDC 60 RPM MOTOR

Powerful gear

motor from

automotive

application.

No-load rating:

60 RPM @ 12VDC, 2.5A.

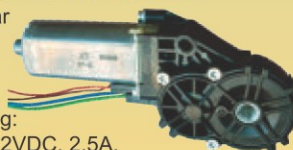
Overall Dimensions: 7.5" x 3.0" x 1.6".

Rotating component of motor is a 3/8" dia.

Nylon bushing with four splines. Can be fitted w/ a shaft from either side.

CAT# DCM-467

\$12.75
each

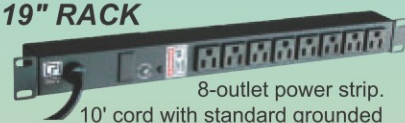


POWER STRIP FOR 19" RACK

8-outlet power strip. 10' cord with standard grounded plug & sockets. 12A 125V 50/60Hz. Power indicator lamp. cULus.

CAT# PST-8409

\$22.00
each



WIRE MANAGEMENT KIT

A real bargain!

Includes 33 Ft

(10 Meters) of heavy

duty 7/8" I.D. black

spiral wrap tubing

and a wire-guide tool

that makes it easy to

wrap the tubing around

already connected cables.

CAT# WMK-1

2 for \$10.00 each

\$15.00
each



Q6025P 25A 600V TRIAC

Teccor Q6025P. Triac.

Fastpak TO-3 base.

CAT# Q6025P

\$5.50
each

10 for \$5.15 each • 50 for \$5.00 each



40 HOOK-UP WIRES, 24" COLOR-CODED, PREPPED

80 Feet of AWG 20

stranded, tinned wire.

Forty 24" lengths. 10

pcs each of black,

white, blue and red.

Most prepped with QC

connectors and pins. A

few stripped and tinned.

Good quality 90°C, 300V

UL, CSA wire. CAT# WH-24

\$3.00
each



In this column, Tim answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. Send all questions and comments to: Q&A@nutsvolts.com.

- Question on R/C Robot Question
- Free to Air Satellite TV
- Testing GFCIs Properly

Post comments on this article at www.nutsvolts.com/index.php?/magazine/article/january2015_QA.

Question about a Previous Q&A Question (R/C Robot)

Q In reference to **Figures 3 and 4** of the November 2013 Q & A (page 22), there were three different answers to my question: (1) Using the Actiontec TV wireless HDMT kit; (2) Using the data channels from a 27 MHz citizen's band (CB) radio; and (3) Using the New Zealand module. I am wishing to go with using the data channels from the 27 MHz CB radio. To do this, I am assuming that I will follow **Figure 4** from the November 2013 issue, with four CB radios, two transmitters, and two receivers.

However, I am confused on how to access these data channels on the citizen's band radios. As a reminder, I am building a robot, using two Raspberry Pis: the first one being the robot's brain; and the second being the base unit, controlling and monitoring the robot, with communication to each other up to 1,000 yards.

Would you please give me an idea of which CB radios would have these data channels, and how to design these two circuits to access the data channels? I am also assuming that I will be needing two CBs for the robot, connecting one as the transmitter to pin 26 (GP-7) and the other as the receiver to pin 5 (GP-3) on the Raspberry Pi, then two more radios for the base unit, connecting one as a transmitter to pin 3 (GP-2) and the other as the receiver to pin 26 (GP-7) on the Pi.

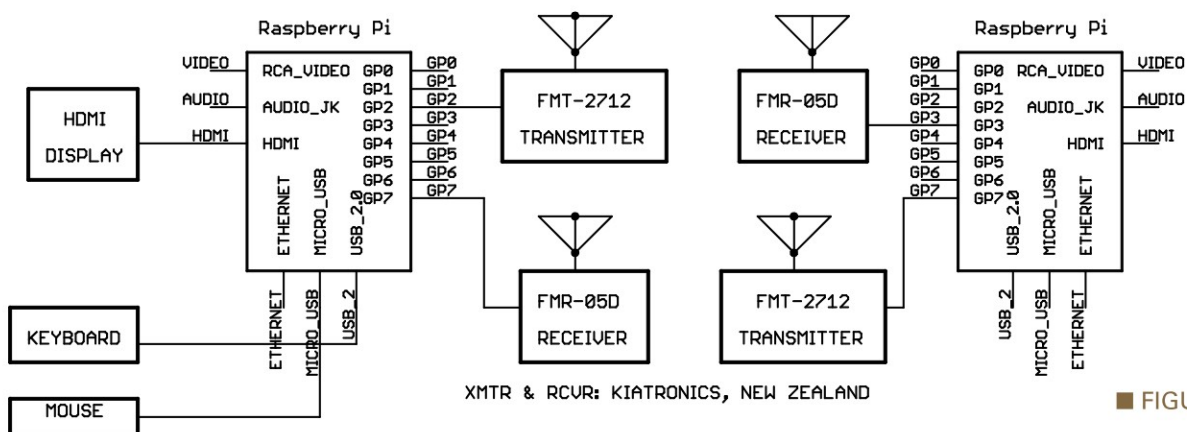
— Timothy Harner, Monticello, UT

A I am including **Figure 4** for reference. The FMT 2712 and FMR 05D are modular FM transmitters and receivers, respectively, operating in the 27 MHz range, made by Elsema for radio control (R/C) hobbyist projects such as aircraft or cars, so using them with a robot is an excellent application. The connections to the Raspberry Pi appear correct since the general-purpose Input/Output (GPIO) can be used with the +5 volt signals to and from the modules. The Pi's GPIO are unbuffered and unprotected, so I would recommend optical isolators to prevent "fried" Pi (pun intended).

This project will require two FMT 2712s and two FMR 05Ds, with each set on a different frequency such as 27.045 and 27.145 MHz for full duplex operation (communications in both directions simultaneously). The "big" deal in this project is to program the Raspberry Pi to perform the command operations.

Also be sure to encode your signals with a code header that identifies your signals exclusively to prevent interference with CB radios, garage door openers, and R/C hobby devices (you may need to experiment with the frequencies based on the local channel usage).

This project has a special appeal to me since I taught robotics at a community college, but only got to work with the industrial models — not the fun stuff like your project. If you are building a flying robot, be sure to check with local and Federal Aviation Administration (FAA) regulations, which at present are very fluid.



■ FIGURE 4.

Here's a trip down memory lane: As an eight year old on a trip through the mountains, I conceived and discussed with my parents a flying vehicle that could be hand launched to show video images of the road on the other side of the mountain. The year was 1958 when computers were bigger than our house, video cameras were weighty, and I had no capabilities to build such a device. Fast forward to 2014 and you can buy such devices for a few hundred dollars.

At 11 years old in 1962, a friend and I "dreamed" of a lightweight glider using the Rogallo wing that we could fly by jumping from a backyard shed. Unfortunately, Barry Palmer had already developed the hang glider a year earlier. Drat, ahead of my time again.

Free to Air Satellite TV

Q Over the years, the *Nuts & Volts* store has advertised a book, *Build Your Own Free to Air Satellite TV System*. I would like to know if it would be possible to design a circuit to convert this free to air satellite TV system to receive and transmit a free to air satellite Internet service.

— Timothy Harner
Monticello, UT

A *Build Your Own Free-to-Air (FTA) Satellite TV System* by Dennis G. Brewer (SKU17182) is still available from the NV webstore for \$30 plus shipping and handling (See <http://store.nutsvolts.com/product.php?productid=17182&cat=347&page=1> to order). The book's description reads:

"Includes step-by-step instructions that show how to select and install all the necessary satellite system components in a home, apartment, office, remote cabin, SUV, or recreational vehicle. Explains how to integrate FTA with existing systems, as well as how to integrate other options such as Over the Air (OTA), video over the Internet, and movies on demand. Shows you how to legally receive hundreds of channels that include programming for music, news, PBS, weather, foreign language television and radio, and talk radio — all for free. Helps you select and

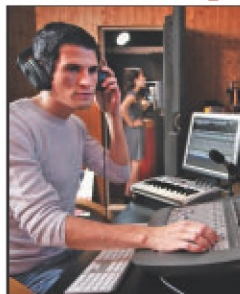
install the cables and connections, switches, receiver, and auxiliary components to get the most value from the system and do it on a very tight budget."

I will have to refer you to Mr. Brewer's book for the "how to" of using FTA satellite over the Internet. I know this sounds like the old grade school book reports ending "and to find out more read the book" to encourage folks to buy it, but the process of FTA is a bit too involved (288 pages in the Brewer book) for the Q&A column. FTA satellite dishes and receivers are available from such sources as www.thesatelliteshop.net/free-to-air-fts-709.html or www.skyvision.com, among many others.

Testing GFCIs Properly

Q I have a question about testing a GFCI. Not being an electrical guy, I would like to know how you know what the GFCI or RCD opens at .5 milliamps or higher. I have assumed the following (I know, never assume), but when you push a test button, doesn't that automatically let voltage go through the GFCI? It should open, you press reset, and the

Make up to \$100 an Hour or More!



Be an FCC LICENSED ELECTRONIC TECHNICIAN!

Get your "FCC Commercial License" with our proven Home-Study Course!

- No costly school. No classes to attend.
- Learn at home. Low cost!
- No previous experience needed!
- GUARANTEED PASS! You get your FCC License or money refunded!



Move to the front of the employment line in Radio-TV, Communications, Avionics, Radar, Maritime and more.

Call for **FREE** info kit: **800-877-8433** ext. 109

or, visit: www.LicenseTraining.com

COMMAND PRODUCTIONS • FCC License Training
Industrial Center, 480 Gate Five Rd., PO Box 3000, Sausalito, CA 94966-3000



system goes back to normal.

So, how do I know it opened at five milliamps? Also, the number of times the GFCI is opened ... doesn't that potentially damage the breaker? If I test it monthly, how often do you recommend changing them out? If there is a way to do this, what equipment do I need? A regular amp/voltage meter? Thanks and best regards.

— William (Bill) Griffin
SSHO Phase III

A GFCI — which stands for Ground Fault Circuit Interrupter — is a device which monitors the flow of current in the "hot" and "neutral" electrical wires in order to sense a ground fault in which the currents in these two wires are not equal (as is the normal case). If the GFCI senses a four to five milliamps (mA) current imbalance, the circuit is broken within 33.3 milliseconds to protect the human.

The ground fault normally encountered is the current running through another conductor (such as the human body) which we try to prevent. Roughly 30 mA of AC or 100 mA of DC current will cause the heart to fibrillate which causes death.

Electricians should ALWAYS work in pairs and know CPR. (Kudos to the power company line technicians who work on energized high voltage lines.)

The GFCI is basically a fast circuit breaker which senses a potential electrocution, so breaks the current to protect human life. GFCIs come in a wide variety, from the receptacle with the "RESET" and "TEST" buttons found in restrooms, to the industrial devices on lines carrying hundreds of amps.

The Residual Current Detector (RCD) is the same thing as a GFCI. [Aside: Years ago, I heard a story of an engineer who specified GFCI breakers on lighting circuits that fed a fluorescent lighting bank. The way fluorescent operated on start-up tripped the GFCIs so these breakers had to be replaced. Expensive lesson!!!].

GFCIs are used in residences in bathrooms, kitchens, swimming pools, hot tubs, spas, etc., where humans, electricity, and water are close together. Your concerns about whether or not the GFCI trips at the proper 5 mA is akin to the old problem of whether or not the light in the fridge goes off when you close the door, but involves human safety.

Residential "fixed" GFCIs should be tested monthly by ensuring that the test button "pops" the GFCI

WORLD'S MOST VERSATILE
CIRCUIT
BOARD HOLDERS

Model 324 Our Circuit Board Holders add versatility & precision to your DIY electronics project. Solder, assemble & organize with ease.

VISIT US ON
Twitter Facebook

MONTHLY CONTEST
Visit us on Facebook® to post a photo of your creative PanaVise project for a chance to win a PanaVise prize package.

PANA VISE®
Innovative Holding Solutions

7540 Colbert Drive • Reno • Nevada 89511 | (800) 759-7335 | www.PanaVise.com

AP CIRCUITS
PCB Fabrication Since 1984

As low as...
\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com

VISA MasterCard PayPal IPC MEMBER ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES BBB

breaker, whereas portable GFCIs on power poles are checked before EACH use.

Testing does eventually wear out the GFCI, but when balanced with safety it is always best to "err" on the side of caution. When I was studying Reliability Centered Maintenance (RCM), one example was the "lamp test" option which only proves that the lamp works when you push the button, and eventually "wears out" the lamp.

GCFI testers such as the Gardener Bender or Amprobe make GCFI circuit testers that plug into the GFCI receptacle. These devices do not give an amp reading but are designed (backed by legal liability induced testing) to ensure the GCFI is safe.

Replace ANY GFCI that fails the test. If this seems too time-consuming, remember the old saying, "Life is short, death is forever."

You could construct a test rig by inserting resistors in the hot and neutral wires of an extension cord, measure the reduced down voltages across the two resistors with a microcontroller (e.g., Arduino, Stamp, PIC), program the controller to calculate the current difference between the legs, then log the data as you push the test button to see if the current difference exceeds 5 mA.

This is a very involved process, however, with tons of

room for errors which could make the measurements invalid and thus pass an unsafe GFCI. Using a commercial tester is the best route. [Note for readers: SSHO stands for Site Safety and Health Officer.] **NV**

Can't figure out that pesky circuit or don't understand the components? Let Tim help!

Send any questions and/or comments to:

Q&A@nutsvolts.com

Start learning electronics with an all-in-one lab!

- * Audio Range Function Generator
- * Audio Range Oscilloscope
- * Spectrogram Function
- * Three Power Supplies
- * Large Breadboard
- * Free Ebook

Explore the world of analog electronics with our all-in-one learning system. The free ebook will teach the basics with labs you can build and verify using our Analog Bakerboard Educational System!



**LumiDax
Electronics LLC**

www.lumidax.com

Phone: 541-510-9264

BEST SCOPES, BEST PRICES



PASSPORT-SIZE PC SCOPES \$162+

Great scopes for field use with laptops. Up to 200MHz bandwidth with 1GSa/s, high speed data streaming to 1MSa/s, built-in 1GSa/s AWG/wfm gen. **PS2200A series**



30MHz SCOPE

\$289

Remarkable 30MHz, 2-ch, 250MS/s sample rate scope. 8-in color TFT-LCD and AutoScale function. Includes FREE carry case and 3 year warranty! **SDS5032E**



50MHz SCOPE

\$399

50MHz, 4-ch scope at 2-ch price! Up to 1GSa/s rate and huge 12Mpts memory! Innovative "UltraVision" technology for real time wfm recording. FREE carry case! **DS1054Z**



60MHz SCOPE

\$349

Best selling 60MHz, 2-ch scope with 500MSa/s rate plus huge 10MSa memory! 8-in color TFT-LCD. Includes FREE carry case and 3 year warranty! **SDS6062V**



70-300MHz SCOPES

\$839+

Fast, versatile 2-ch 2GSa/s scopes with 8-in WVGA LCD, integrated generator, 14Mpt memory, very low noise floor. FREE carry case available! **DS2000A series**

FREE TECH SUPPORT - GREAT CUSTOMER SERVICE



Harnessing the Power of USB

For a while now, I've been requesting reader feedback about the future direction of PICAXE Primer articles. At this point, I have received almost 40 responses, and I want to thank all the readers who took the time to email me their preferences. Here's a quick summary of the results:

- 57% want the Primer to return to "pure PICAXE" projects
- 37% would prefer to continue our PICAXE-Pi explorations
- 06% said either option would be fine with them

The "Pure PICAXE" readers are clearly in the majority, so that's the direction we will take — at least for the immediate future.

This month, we're going to construct a couple of simple USB-based power supplies for our PICAXE projects. I've wanted to explore this topic for a while, for two reasons. First, my main computer is a laptop and I often need to work on a PICAXE project in more than one location, so the less hardware I have to lug around, the better! That's the main reason I originally developed the USB-based AxMate programming adapters that we discussed back in the June 2010 Primer.

However, sometimes I need to work on a project that involves the AXE027 PICAXE USB programming adapter. In that situation, I also have to carry the 12V wall wart supply that I use with my BBP-01 breadboard power supply. A USB-based power supply would enable me to leave the bulky wall wart at home, and just carry a lightweight USB cable that I can plug directly into my laptop or my cell phone's USB charger, which I always carry anyway.

The second reason behind this month's project is that I have a couple of USB-related projects already "in the queue" for upcoming Primer articles. (Don't you just love the suspense?) This month's Primer will give us a jump-start in that direction, and in the direction of a little more surface-mount soldering in future articles if you're up for a challenge!

Before we begin to construct our first USB power supply, I do want to give everyone a brief RazzPi update.

The three boards we discussed in the previous Primer are (finally!) available on my website. If you're a reader who is also interested in the Raspberry Pi, you may want to check them out.

As an aside, I recently succumbed to an irresistible Best Buy sale and purchased a Pebble smart watch. I'm very impressed with how easy it is to set up a PICAXE-Pi project that automatically sends SMS text messages to the Pebble, providing me with real time data updates on my watch. When I have more experience with my "P-3" (PICAXE-Pi-Pebble) project, I'll add the information to my website, and/or write a non-Primer article about it.

A Brief Introduction to USB Power Specifications

Before we embark on our first USB power supply project, we need to discuss a couple of the USB power specifications. USB3 devices have been available for about five years now, but many of us are still using USB2 hubs and cables, so we need to consider both versions.

Figure 1 presents a summary of the differences between the USB2 and USB3 specifications that are relevant to our power supply project. As you can see, USB3 is able to supply significantly greater power to connected devices. However, the distinction between high and low

■ FIGURE 1.
Selected
USB power
specifications.

	USB2	USB3
Max. Current for Low-Power Devices	100mA	150mA
Max. Current for High-Power Devices	500mA	900mA
Minimum Voltage Supplied	4.75V	4.45V
Maximum Voltage Supplied	5.25V	5.25V

power devices requires clarification.

Whenever a device is plugged into a USB port, it's initially considered to be a low power device. If the device requires more current than is initially provided, it's the responsibility of the device to communicate with the USB port, and specify the amount of current that it needs. In other words, all high power USB devices must be intelligent. Our USB power supply does not have that capability, so we're limited to 100 mA for USB2, and 150 mA for USB3.

That limitation may seem significant for a power supply, but I've been using the AxMate programming adapters with my PICAXE projects for more than five years, and have yet to encounter a problem. Also, if a project does require a high power supply, we always have the option of using a USB wall wart, and connecting the power supply cable directly to the wall wart rather than a USB port.

On the other hand, there is an advantage to using a USB port for the power supply. If we were to accidentally create a short circuit on a project, the USB port on a computer or an external hub would immediately shut down the power to the project, so we would be much less likely to have damaged any components.

Now, let's turn our attention to the voltage specifications listed in **Figure 1**. The most important one is that the voltage supplied by all USB2 and USB3 ports, hubs, and wall wart supplies must be at or below a maximum of 5.25V. Since the maximum operating voltage for all current PICAXE processors is 5.5V, the USB specifications are well below the safe maximum voltage for PICAXE processors.

Just to check those figures, I measured the voltage provided by four different USB2 wall warts that I had on hand (and one USB2 hub); they all measured between 5.05V and 5.22V. I don't know why the minimum voltage was lowered for

the USB3 specifications, but the voltage provided by the USB3 ports on my laptop measures at 5.21V, so I don't think the USB3 minimum voltage will be a problem.

USB Connectors

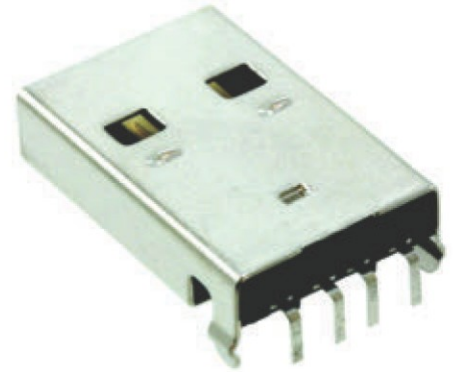
As you know, there are a variety of USB connectors on the market: standard, mini, and micro; male and female; type A and type B, etc. For the stripboard power supply circuits that we'll implement this month, we will use the standard male and female type A connectors because they both have a single row of four pins that are very "stripboard friendly" (see **Figure 2**).

If you look closely at the connectors in **Figure 2**, you can see that the pin spacing isn't exactly uniform, but the pins are very easy to coax into the holes of four adjacent traces on a stripboard. In addition, the two larger case "pins" on each connector can be straightened easily, and they will fit into two enlarged holes on a stripboard.

Figure 3 shows the pin-outs of the male and female type A connectors. At first glance, the pin order may seem reversed, but don't forget that we're looking into each connector. So (for example), when a male connector on the end of a cable is inserted into the female connector on a device, the two connectors are facing each other, and the pins do line up correctly.

For our USB power supply, we're only interested in pins 1 (+5V) and 4 (ground), so the easiest (and safest) thing to do is to snip the middle two data pins from the connector before installing it in a stripboard circuit.

We're going to run through the details of constructing two different power supplies: one with



■ **FIGURE 2.** Standard USB male type A connectors.

a male connector, and the other with a female connector. Both supplies are functionally identical, so you will most likely choose to construct only one of them. Of course, the "male" power supply requires a male-female USB cable in order to connect it to a USB power source (computer, hub, wall wart, etc.), while the "female" power supply requires a male-male USB cable for its connection.

If you only have one of those two types of USB cables available, that may strongly influence your decision, but I should warn you in advance — the female power supply is a little more difficult to construct, and both types of USB cables are readily available on eBay (from US sellers) for less than \$2, so I wouldn't blame you if you chose the easy way, and just constructed the male power supply.

Before we actually begin construction of our power supplies,



USB A Male

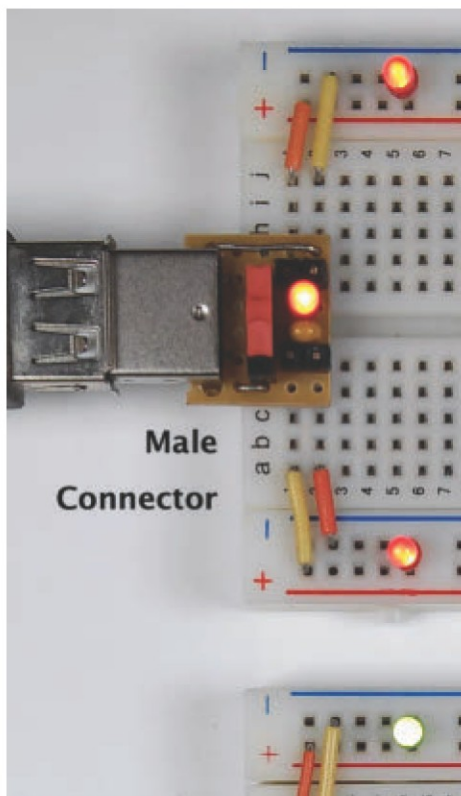


USB A Female

[Each image is looking into the connector.]

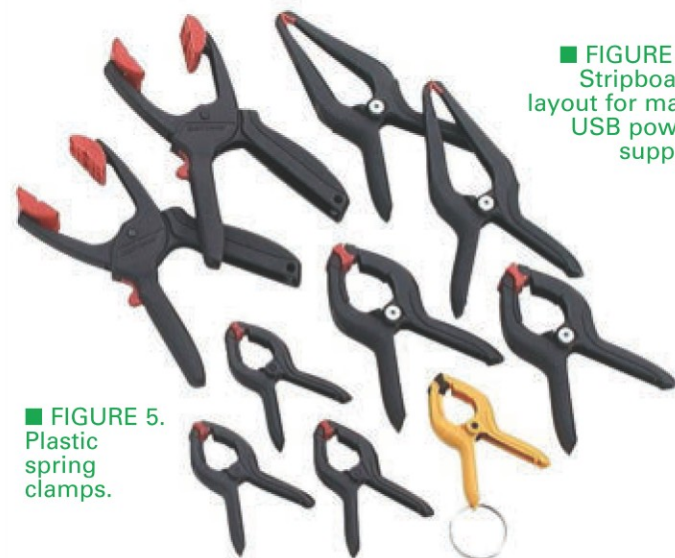
Pin	Name	Signal
1	Vcc	+5V
2	D -	Data -
3	D +	Data +
4	GND	Ground

■ **FIGURE 3.** Standard USB type A pin-outs.



■ FIGURE 4. Completed USB power supplies.

there are two more points that I want to mention. First, both supplies are very small (see **Figure 4**), which adds to the challenge of soldering the parts in place. In addition to a small vise, a set of small plastic spring clamps can be very helpful when working with small boards and parts.



■ FIGURE 5. Plastic spring clamps.

I have a 10-piece “Clamp Hobby Set” (Model# 31593) that I found at www.sears.com (see **Figure 5**) which I tend to use whenever I’m assembling a stripboard or printed circuit board (PCB) circuit; long needle-nose clamps and the small flat faced clamps were especially helpful in this month’s project.

The second point relates to the long-standing “RTFM” maxim. (If you’re not familiar with that one, Google is your friend!) For several years, I’ve been searching unsuccessfully for a small power switch that can handle more current than the one that has long been available on my website. I recently found one that has a footprint that’s slightly smaller than 0.1 inches by 0.4 inches, and it can handle as much as 500 mA of current. (Its red top is visible in the **Figure 4** photo.)

I was eager to use this switch in the USB power supply project, but when I built my first power supply, it didn’t work; power was supplied to the breadboard regardless of the position of the switch! At first, I thought the switch was simply defective, but I finally figured out that it works “backwards.”

Every small slide switch I’ve

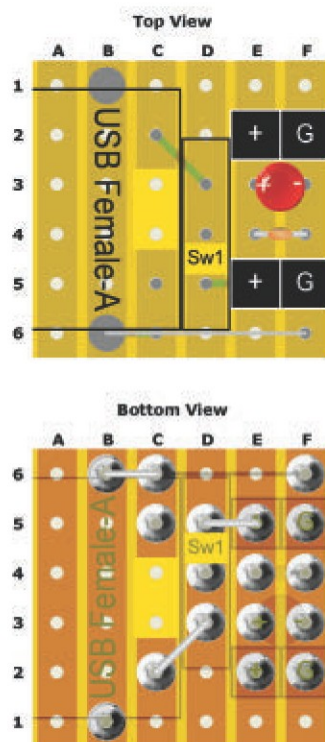
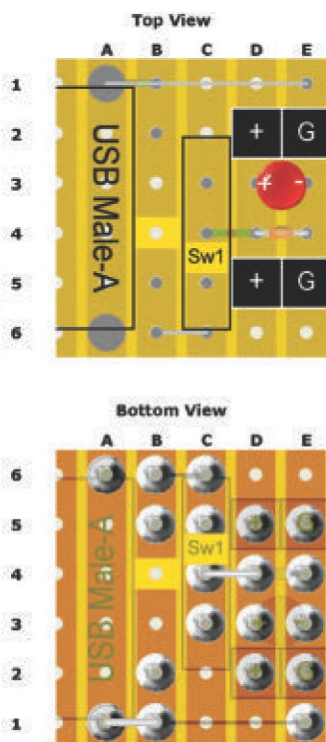
used in the past works the same way: Slide the lever up, and the two upper contacts on the switch are electrically connected; slide the lever down, and the two lower contacts are electrically connected. With my nice new switch, the opposite is true: lever-up connects the lower two contacts, and lever-down connects the upper two contacts! Of course, I did ultimately read the one page “manual” where that fact is clearly stated.

I didn’t relate that sad little story simply to confess my foolishness; I wanted to warn you that, if you decide to use a power switch other than the “reversed” switch that’s available on my website, you will need to modify both stripboard layouts in order to get a “normal” switch to function correctly. With that in mind, let’s move on to actually build a USB breadboard power supply.

Constructing a USB Male Power Supply

The circuit for both USB power supplies is so simple that neither a schematic nor a separate parts list is

■ FIGURE 6. Stripboard layout for male USB power supply.



■ FIGURE 7. Stripboard layout for female USB power supply.

required. The top and bottom stripboard layout — which is shown in **Figure 6** — is all we really need. (Large versions of the layouts for both of this month's power supplies are available for downloading at the article link).

In the layout for the male power supply, the connector's +5V line (pin 1) is located near the bottom of the board at B5. The short B6-C6 jumper on the bottom of the board connects the +5V power line to pin C5 on the switch. As a result, when we move the lever of our "reversed" switch to the "up" position in the layout, the switch electrically connects C4 and C5, and the short C4-D4 jumper on the bottom of the board connects the +5V power to the two power pins at D2 and D5.

The only other components on the board are a .01 μ F decoupling capacitor and a 3 mm resistorized LED (which are both placed between the +5V and ground output traces), and the two two-pin headers (which are the longer reverse-mounting type that we usually use to connect a stripboard circuit to a breadboard). All the necessary parts are available on my website, but you may already have most of them on hand.

In **Figure 6**, note the half-trace on the left side of the stripboard. If you refer back to **Figure 2**, you will see a small notch in the case of the male USB connector, just above the large "pin" on the connector's case. In order to be able to fully insert the connector into a stripboard, the stripboard cannot extend beyond that notch. A half-trace is about all that will fit underneath the male connector that's available on my site. (If you use your own USB connector, check to see how far the stripboard can extend underneath it before cutting it to size.)

Also, note that the half-trace is **not** included in the column labels. As usual, read through the complete list of assembly instructions that follows to be sure you understand the entire procedure before

beginning to assemble the board:

- Cut a piece of stripboard to size (5-1/2 traces with six holes each).
- Cut traces B and C as shown in the layout.
- Using a 5/64" drill bit, enlarge the two holes at A1 and A6. Drill both holes from the bottom of the stripboard to avoid tearing out excess copper from trace A.
- Install the B6-C6 jumper, then solder both leads on the bottom of the board.
- Use pliers to straighten the two large pins on the connector's case.
- Snip the two data pins (pins 2 and 3) from the connector.
- Use a stripboard chisel to remove the two small plastic nubs on the bottom of the USB connector.
- Insert the male USB connector firmly onto the stripboard. Clamp it firmly in place and make sure it's parallel to the board. Do **not** solder it in place yet.
- Cut a piece of wire for the A1-E1 jumper, allowing a little extra wire on the A1 end so that the jumper can be bent back on the bottom of the stripboard, as shown in the bottom view of the layout.
- Install the jumper, bend it back from A1 to B1 on the bottom of the board, and snip the excess length so that it doesn't extend beyond trace B.
- Solder the jumper at A1, B1, and E1. At A1, heat the pin on the case of the USB connector for a few seconds before applying solder.
- Solder the other USB case pin at A6. Again, preheat the pin.
- Install the capacitor. Only solder the ground lead at E4; do **not** solder or trim the positive lead at D4 yet.
- Install the resistorized LED (observe polarity). Solder and snip both pins.
- Install the switch (no polarity). Solder the pins at C3 and C5. Do not solder the pin at C4 **yet**.
- Bend the capacitor lead from D4 to C4. Use a small screwdriver to press it flush against the bottom of

the board so that it touches the middle pin of the switch at C4. Snip the excess lead so that it does **not** extend past trace C, and solder the leads at D4 and C4.

- Snip all pins and excess leads on the bottom of the board as close as possible.
- Sand and clean the bottom of the board.
- Snip off the short ends of the pins from the two reverse-mounting two-pin headers.
- Insert one header from the top of the board at D2 and E2. (You may need to bend the LED slightly so the header sits flat on top of the board.)
- Turn the board over, hold it in place with a plastic clamp, support the header with a small piece of scrap wood, and solder it in place (using a minimal amount of solder).
- Insert the other header from the top of the board at D5 and E5, and solder it in place using the same approach as the previous instructions.
- Insert the completed power supply onto a breadboard as shown in **Figure 4**. If any of the solder connections prevent it from sitting flat on the breadboard, remove the power supply and further trim or file those connections until the board can be seated properly on the breadboard.
- Again clean the bottom of the board to get rid of any possible sanding/filing residue.

When you have completed the USB male power supply, test it with the USB power sources you intend to use, and measure the voltage level to make sure it's less than 5.25V.

Constructing a USB Female Power Supply

The female version of our USB power supply project is very similar to the male version that we just discussed, but there are two differences that are important to keep in mind. First, if you look at the stripboard layout in **Figure 7**, you can

see that the female USB connector's +5V line (pin 1) is located near the top of the layout at C2 (rather than near the bottom, as it was in the male version).

The short C2-D3 jumper on the bottom of the board connects the +5V power line to pin D3 on the switch. As a result, when we move the lever of our "reversed" switch to the "up" position in the layout, the switch electrically connects pins D4 and D5. Therefore, the short D5-E5 jumper on the bottom of the board connects the +5V power to the two power pins at E2 and E5.

The second difference is that we're using six full traces this time (because they fit under the female connector), so all six traces are labeled in the layout. Again, all the necessary parts are available on my website, but you may already have what you need. As before, read through the complete list of assembly instructions that follows to be sure you understand the entire procedure before beginning to assemble the board:

- Cut a piece of stripboard to size (six traces with six holes each).
- Cut traces C and D as shown in the layout.
- Using a 5/64" drill bit, enlarge the two holes at B1 and B6. Drill both holes from the bottom of the stripboard to avoid tearing out excess copper from trace A.
- Use pliers to straighten the two large "pins" on the connector case.
- Snip the two data pins (pins 2

and 3) from the connector.

- Insert the female USB connector firmly onto the stripboard. Clamp it firmly in place and make sure it's parallel to the board. Do **not** solder it in place yet.

- Cut a piece of wire for the B6-F6 jumper, allowing a little extra on the B6 end so the jumper can be bent back on the bottom of the stripboard, as shown in the bottom view of the layout.

- Install the jumper, bend it back from B6 to C6 on the bottom of the board, and snip the excess length so that it doesn't extend beyond trace C.

- Solder the jumper at B6, C6, and F6. At B6, heat the pin on the case of the USB connector for a few seconds before applying solder.

- Solder the other USB case pin at B1. Again, preheat the pin.

At this point, we're ready to install the switch, but there's a complication that we need to address before doing so. The top of the switch is lower than the top of the female USB connector, making it difficult to access. I'll describe how I solved the problem, but if you find a simpler solution, go for it!

I first tried to raise the switch by snapping a three-pin piece from a long male header, pulling the three pins out of the black plastic, and then inserting the black plastic onto the pins of the switch. That raised the top of the switch above the top of the female USB header, but it didn't leave enough pin length on the switch to enable it to be soldered

onto the stripboard.

Next, I removed the black plastic from the switch pins, held it firmly with a pair of pliers so the three holes were on the exposed surface, and the jaws of the pliers were positioned so that about half the height of the plastic was exposed.

Then, I used my bench-top disc sander to (carefully!) remove a little less than half the height of the black plastic. In other words, the black plastic — which is normally 0.1" in height — ended up being about 0.06" in height. When I inserted the plastic back onto the pins of the switch and installed the switch back onto the stripboard, there was just enough pin length on the bottom of the stripboard to enable me to solder the switch in place.

If this little complication hasn't discouraged you, let's continue:

- Insert the modified switch (no polarity) onto the board as shown in the layout, and solder **only** the middle pin at D4. Do not solder the pins at D3 or D5 yet.

- Cut a 3/4" piece of thin jumper wire (e.g., a lead from a 1/6W resistor), bend it in half at its center, and use small needle-nose pliers to crimp it onto the bottom of the board as shown in **Figure 8**.

- Snip the excess leads so they do not extend beyond trace C, and then solder the jumper in place at C2 and D3.

- Install the capacitor as shown in the layout (no polarity).

- Solder its leads at E4 and F4.

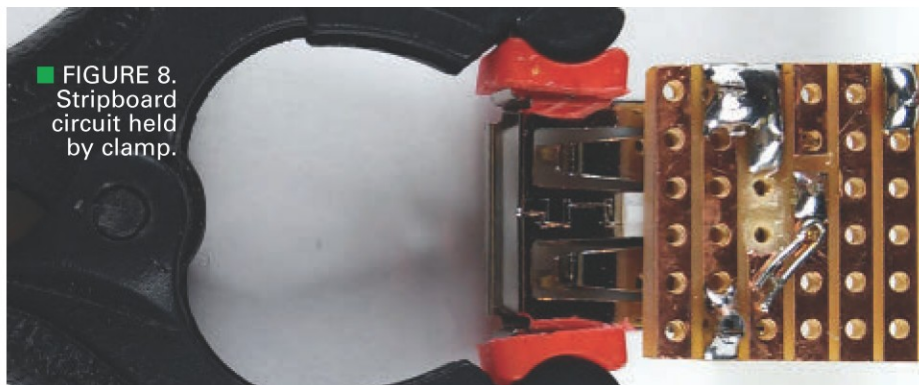
- Install the resistorized LED as shown in the layout (observe polarity).

- Solder its leads at E3 and F3.

- Snip all pins and excess leads on the bottom of the board as close as possible. (Do not yet snip the unsoldered switch pin at D5.)

- Sand and clean the bottom of the board.

- Snip off the short ends of the pins from the two reverse-mounting two-pin headers.



■ **FIGURE 8.**
Stripboard
circuit held
by clamp.

- Insert one header from the top of the board at E2 and F2. (You may need to bend the LED slightly, so that the header sits flat on top of the board.)

- Turn the board over, hold it in place with a plastic clamp, support the header with a small piece of scrap wood, and solder it in place (using a minimal amount of solder).

- Insert the other two-pin header from the top of the board at E5 and F5, and solder the pin at F5 in place using the same approach as the previous instructions. Do **not** solder the pin at E5 yet.

- Cut a 1/2" piece of thin jumper wire (e.g., a lead from a 1/6W resistor), and use small needle-nose pliers to form a U-shaped bend at one end.

- Crimp the U-shaped end onto the switch pin at D5, and trim the wire so that it just touches the header pin at E5.

- Use a small screwdriver to press the wire flat against the bottom of the board, and solder it in place at D5 and E5.

- Trim the soldered connection at D5 as close as possible.

- Insert the completed power supply onto a breadboard as shown in **Figure 4**. If any of the solder connections prevent it from sitting flat on the breadboard, remove the supply and further trim or file those connections until the board can be seated properly on the breadboard.

- Again, clean the bottom of the board of any possible residue.

When you have completed the USB female power supply, test it with the USB power sources you intend to use, and measure the voltage level to make sure it's less than 5.25V.

Developing a Micro-B USB Power Supply

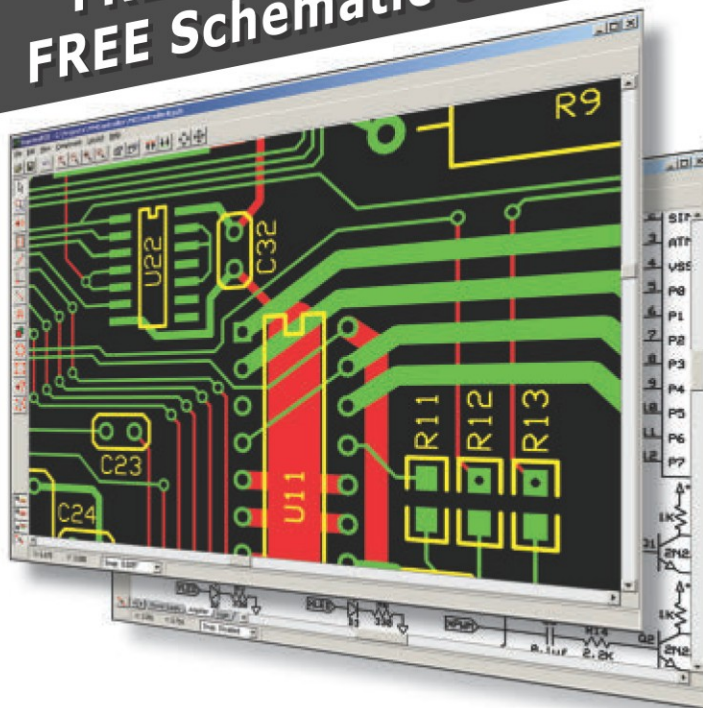
As we saw earlier in **Figure 4**, both versions of our USB power supply are very small, but I couldn't resist the temptation to develop an even smaller supply. The type A

connectors that we used in both stripboard projects are, by far, the largest components on the boards, so the obvious way to shrink the power supply is to use a smaller connector.

At this point in time, micro-USB connectors are just about universal; almost all current cell phones and cameras include a female micro-USB

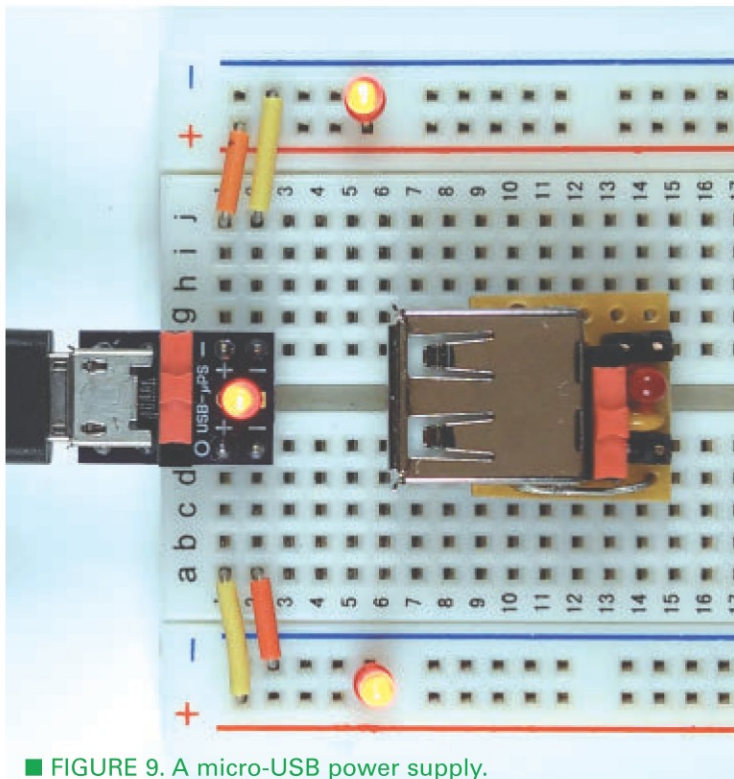
type B connector, so that's the one I chose. However, micro-USB connectors are far from "stripboard friendly" because they are surface-mount devices, and their tight pin spacing (0.026 inch) makes them somewhat difficult (but not impossible) to hand-solder. With that in mind, I developed a PCB version

\$51^{For 3} PCBs
FREE Layout Software!
FREE Schematic Software!



- 01 DOWNLOAD our free CAD software
- 02 DESIGN your two or four layer PC board
- 03 SEND us your design with just a click
- 04 RECEIVE top quality boards in just days

expresspcb.com



■ FIGURE 9. A micro-USB power supply.

of a micro-USB power supply which is shown in **Figure 9**. As you can see, it's considerably smaller than our stripboard power supply.

If you read the June 2013 Primer column, you know that SOIC devices have a 0.05 inch pin spacing, which makes them relatively easy to solder by hand using the traditional "pin by pin" approach. In comparison, a micro-USB connector's pin spacing is roughly twice as tight as that of a SOIC device, and I doubt whether anyone could hand-solder a micro-USB connector using the pin by pin approach.

However, with a little practice, a second method of soldering by hand (called drag-soldering) does make it possible to hand-solder micro-USB connectors. If you're curious, search on line for "drag soldering," and watch a few of the thousands of videos that are available.

If you're feeling adventurous, the PCB for my micro-USB power supply as well as the necessary parts and documentation are available on my website. Even if you're not interested in learning to drag-solder, there will definitely be a couple of SMD projects in future Primer articles, so you may want to have some fun working on your SMD soldering skills in the next few weeks. **NV**

SUPERIOR **EMBEDDED** SOLUTIONS



DESIGN YOUR SOLUTION TODAY
CALL 480-837-5200

www.embeddedARM.com

TS-7670 and TS-7680 Industrial Computers

- Up to 454 MHz ARM w/ 256 MB RAM
- 2 GB Flash Storage
- Industrial Temperature (-40 °C to 85 °C)
- DIO, CAN, Modbus, RS-485

TS-7670 Features:

- GPS and Cell Modem
- 1x Ethernet
- 2x microSD Card Sockets

Pricing
starts at
\$129
Qty 100

\$168
Qty 1



Low cost plastic
enclosure available

NEW! TS-4900 High Performance Computer Module

- Up to 1 GHz Quad Core ARM CPU
- Up to 2 GB DDR3 RAM
- WiFi and Bluetooth
- 4 GB Flash and microSD
- Gigabit Ethernet
- SATA II and PCI-Express
- DIO, CAN, COM, I2C, I2S
- Industrial Temperature (-40 °C to 85 °C)
- Supports Linux & QNX
- Android & Windows Coming Soon

Pricing
starts at
\$99
Qty 100
\$134
Qty 1



**COMING
SOON!**

TS-7970
SBC Version
of the TS-4900



We've never
discontinued a
product in 30 years!



Embedded
systems that are
built to endure



Support every step
of the way with
open source vision



Unique embedded
solutions add value
for our customers

An Intro and Antennas

Welcome to the world of wireless know-how in the form of amateur or "ham" radio. Where else can you be an electronics and programming whiz, study solar and atmospheric phenomena, design your own communication system, and provide valuable public service — all at the same time? Amateur radio and the *Nuts & Volts* readership have a lot in common. Let's get to know each other!

I'm thrilled to help return ham radio to the pages of *Nuts & Volts*! In every other issue, I'll be discussing some aspect of ham radio technology that you can use on your workbench and in your projects — whether you have (or get!) a license or not.

Over the years, *NV* has featured ham radio in articles and columns so ham radio was never truly absent. The magazine's editor is a ham (NU1N) and Paul Verhage — maven of the high altitudes and Near Space columnist — is also known as KD4STH.

Many of the authors hold an amateur "ticket," so maybe they will share their call signs with us in future articles. You may be surprised at how widespread amateur radio really is!

A little about my background: I have a degree in Electrical Engineering and spent 20 years in various types of industrial and medical product development — both hardware and software. I've had an amateur radio license since my high school days and am now known on the ham bands as NØAX (the slash is silent). For the last dozen years or so, I've been writing and editing books and columns for the American Radio Relay League (www.arrl.org), such as the three licensing study guides; a classic reference for radio technology, *The ARRL Handbook*; and a near-classic antenna reference, *The ARRL Antenna Book*.

Some of my other books and columns are included in the sidebar on resources, including *Ham Radio for Dummies*. I'm pretty active on the air and like to operate in competitive events known as *radiosport*, as well as provide public service and study radio wave propagation. There is more on my "ham radio bucket list" than I will ever get to!

So, what is this ham radio stuff anyway and why should you care? First, there is far, far more to amateur radio in the 21st century than in the movies. Those images you see of glowing tubes and racks full of black crinkle-finish equipment with the jumping meters and dials? They are as obsolete as 8" single-sided floppy disk drives and 7400 family TTL logic! Sure, some of that gear is still out there on the air, but today's ham

radio is up to date and innovative.

Hams are big players in the Arduino and Raspberry Pi communities, just as they are in developing over-the-air digital communications protocols and networks. Even if you're not really interested in the full ham radio experience, you might be interested in using non-licensed wireless data links in your projects, for example. Whatever your specialty, learning about radio will help you select, apply, and use wireless technologies better.

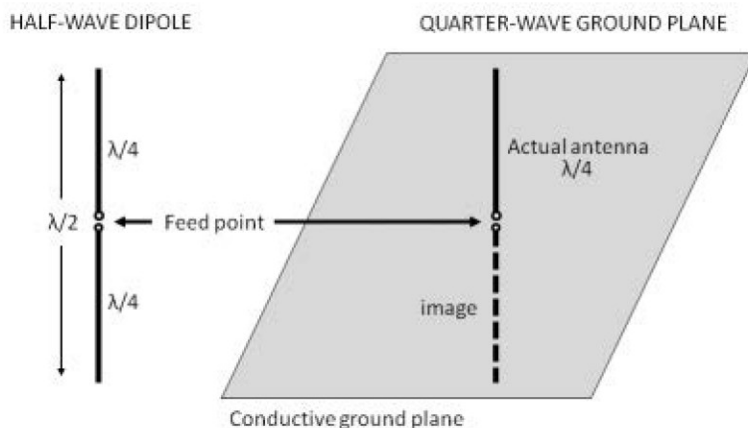
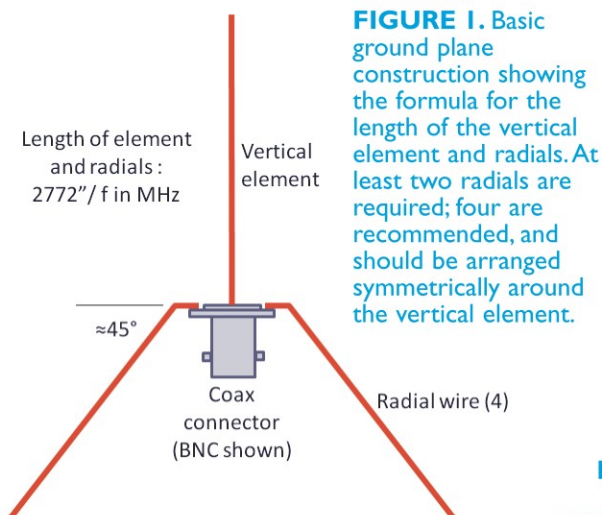
Modern-day ham radio is really a combination of three important components. The first is science: Hams learn about radio circuits and systems, antennas, how signals propagate from place to place, and the antennas that make it happen. The second is skill: By practicing effective operating, hams apply that science to insure that signals get from point A to point B. Finally, the ham combines the science and skill in service of his or her fellow citizens. You may have seen the motto "When All Else Fails," which refers to the ham's storied ability to fill in when commercial and government communications are disrupted.

All three aspects — science, skill, and service — are important, and there is a home for you in whichever area is most interesting.

This column will touch a lot of

Where Does the Term "Ham" Come From?

Everybody wants to know why it's called "ham" radio. While there are many answers floating around, the truth is that no one really has the definitive answer. Nevertheless, after being asked thousands of times, the most common and reasonable source of "ham" is that it was originally a not-very-complimentary term used to refer to the amateurs by commercial and military operators. In those days of spark transmission — the original ultra-wideband signal! — everybody had to share all of the radio spectrum, so interference was a huge problem. The amateurs turned the term into a badge of pride that persists to the present day. It's not an abbreviation for anything, so it's never capitalized. It's referring to the original hackers — the hams.



Your Go-To Source for Radio Know-How – the ARRL

The world's oldest amateur radio organization is also the United States' national amateur radio institution: the American Radio Relay League, or the ARRL (www.arrl.org). Usually referred to by hams as “the League,” the ARRL's motto is “Of, By, and For the Amateur.” It represents amateur radio internationally and in Washington, D.C. to insure that hams have the necessary spectrum to fulfill the amateur service's role.

Within the amateur radio community — more than 700,000 in the United States alone — the ARRL's role is to educate amateurs, support the volunteers who perform all licensing activities, and provide training and service activities to keep the skills of hams sharp and ready.

The ARRL publishes an enormous amount of technical and operating material in whatever area of amateur radio is most attractive to you. If you would like to explore some of the technology resources on the ARRL website, there is a special portal just for you at www.arrl.org/tech-portal.



bases: antennas, transmission lines, batteries, digital protocols, radio frequency (RF) circuits and techniques, test equipment, and the list goes on. We'll discover that components act a lot differently above a few megahertz (MHz) than they do at audio and DC. I will show you how to install those pesky feed line connectors so the signal goes to the right place.

Similarly, we'll take a look at ways to keep RF signals from leaking out of

and getting into your equipment. In some columns (like the one this month), there will be an experiment or activity you can do to gain valuable experience and maybe even a useful gadget. Ready to get started? I thought so!

The Ground Plane Antenna

There is no subject better suited to kick off a column about ham radio technology than antennas. All forms and specialties of ham radio share antennas as a common part of the station. If it's ham radio, you can be sure of an antenna being involved. Actually, a lot of non-amateur electronics also deal with antennas, such as wireless links and mobile phones.

In this column, I'll introduce you to one of the simplest antennas and show you how to make one to use at home — or even design your own.

Figure 1 shows the basic idea: A vertical wire is attached to the center conductor of a coaxial cable connector, and several *radial* wires are attached to the mounting holes of the connector. This particular style of connector is called a *receptacle* or *panel jack* because it is designed to mount on a panel and have a cable attached to it. (*Bulkhead* receptacles

mount with a nut and lock washer in a single hole, and won't work for this particular antenna.)

A *family* of connectors shares common attachment mechanisms and body sizes. The receptacle we're using here is from the BNC family. (Other common connector families include SMA, N, and UHF.)

The vertical wire — called an *element* — is the main part of the antenna that receives the signal. Does the orientation of the element matter? Yes, it does. Radio waves are made up of an electric or *E field* and a magnetic or *H field* that are at right angles to each other.

The magnetic field makes electrons move in tight little circles which is not very useful in creating currents that flow to a receiver. The electric field, however, makes electrons move in a straight line. In this case, the antenna is designed so the *E field* will make the electrons move back and forth along the wire element, into the coaxial cable, and down the cable to the receiver.

The orientation of the electric field determines the radio wave's *polarization* — horizontal or vertical.

When the antenna element and radio wave *E field* are aligned, the antenna receives the strongest signal. The orientation of the antenna's element or elements determines the

Radio and Radiation

The word “radiation” tends to get people excited because the same word is used to apply to all types of radiated energy — from radio waves to infrared light to X-rays and atomic particles. Radio waves as we are discussing here are non-ionizing, meaning they do not have enough energy to knock electrons away from atoms creating ions. The only effect from radio waves on humans is thermal or heating. For radiation to become ionizing, it takes extreme ultra-violet, X-rays, gamma rays, or particles to create ions and cause chemical changes in cells. For more information on safety issues associated with radio waves or RF, see www.arrl.org/rf-radiation-and-electromagnetic-field-safety.

antenna’s polarization. In this case, the antenna is vertically polarized. Since the NOAA Weather Station broadcasts that this antenna is designed to receive are transmitted by a vertically polarized antenna, our antenna should be vertically polarized to receive the maximum signal. (Misalignment is called *cross-polarization* and can result in up to 99% or 20 decibels [dB] of signal loss because the E field no longer makes electrons move along the antenna element as receivable current.)

An Electrical Mirror

The name “ground plane” comes from the four radial wires — so-called because they extend radially from the center. The ground plane acts as an electrical mirror to create an electrical image of the antenna’s missing half. Missing half? Yes, the ground plane antenna is actually one-half of a *dipole* as shown in **Figure 2**.

The mirroring effect of the ground plane is the same as that of the missing part of the dipole. In this case, four radials are enough to do the job. For ground plane antennas mounted on vehicles, the radials are replaced by the sheet metal of a roof or trunk.

A one-half wavelength long dipole is an effective antenna that radiates and receives equally well in all directions around the antenna’s axis. If the dipole is vertical, the equal response from any direction makes it *omnidirectional*.

That omnidirectional response is not repeated if one looks at

the antenna’s response from the side.

Figure 3A shows the ground plane’s *elevation pattern* starting at the horizon (0°) on one side, going over the top of the antenna (90°), and back to the opposite horizon. The antenna receives very little signal overhead because the E field of a radio wave coming from that direction only moves an electron back and forth across the element, not along the element where it becomes a current that can be received.

The notch in the antenna’s pattern is called a *null*, whereas the region of maximum response at the horizon is a *lobe*. An *azimuth pattern* would look down on the antenna from the top and show how well the antenna receives in all directions or azimuths. Since the antenna is omnidirectional, that pattern would be a simple circle.

Figure 3B shows what the antenna’s three-dimensional response looks like — sort of a bagel shape if the antenna was stuck through the middle.

By replacing the lower half of the dipole with its image, the *quarter-wave ground plane* achieves the same

effect of a vertical dipole, but in a smaller package with easier mounting. Ground plane

antennas are common at frequencies from the AM and long-wave broadcast bands (at and below 1.7 MHz), all the way up to about 1 GHz wherever an omnidirectional response is desired.

A Ground Plane Antenna for NOAA Weather Stations

Many scanners and VHF radios have the ability to receive the seven NOAA weather station channels near 162.5 MHz (www.nws.noaa.gov/os/marine/wxradio.htm). The flexible whip (also known as “rubber duck”) antennas provided with portable radios are not very efficient. If you are in an area of weak coverage or are traveling to a remote area, you may need a full size antenna to pull in these stations. By building this simple ground plane antenna, you will be able to receive more of the stations over a wider area.

Parts List and Instructions

- Six foot BNC-to-BNC coaxial cable (RG-58 or RG-8X cable)
- BNC panel jack (UG-290 style or any flange-mount style)
- Eight feet of #14 AWG solid copper wire or brass rod (#12

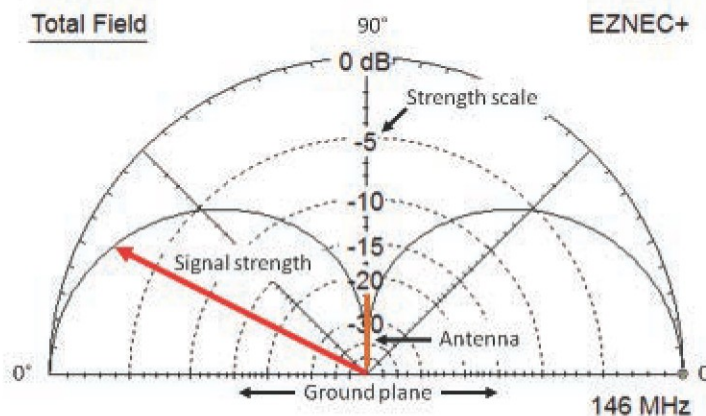


FIGURE 3A. These are radiation patterns showing how strongly an antenna responds to signals arriving from various directions. The distance from the center to the solid line shows the strength of the response in decibels with respect to the maximum response. This figure shows an elevation pattern looking at the antenna from the side.

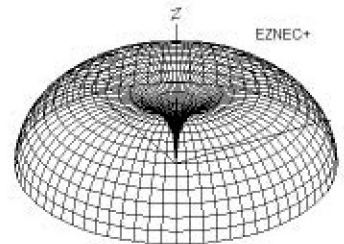


FIGURE 3B. The three-dimensional pattern resembling a bagel with the antenna at the center. The patterns were generated by the EZNEC antenna modeling software (www.eznec.com).

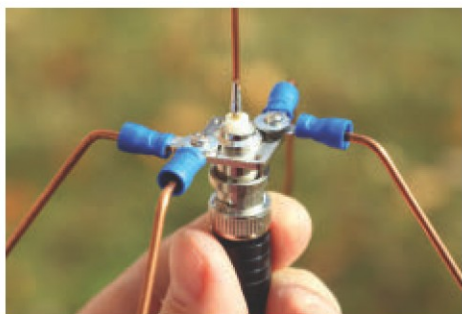


FIGURE 4. Close-up of the assembled ground plane showing one method of attaching radial wires to the BNC connector flange. Radials may be attached permanently by soldering, or temporarily with screws.

to #16 will work and stand up to handling)

- Four ring crimp terminals for 12-16 ga wire (blue insulation) for #4 stud
- #4 hardware to attach terminals to connector flange

Start by calculating the length of wire for the vertical element and the

four radials. All five will be the same length. Use this equation with a frequency of 162.5 MHz:

$$L \text{ (inches)} = 2772 / f \text{ (MHz)} = 2772 / 162.5 = 17.1 \text{ inches}$$

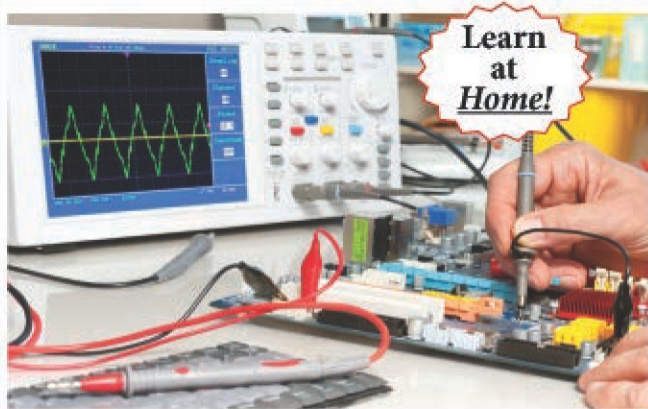
Where does the equation come from? Remember that the ground plane's vertical element is one-quarter wavelength long or $\lambda/4$, where the Greek letter λ stands for wavelength. In free space at the speed of light, $\lambda = 300 \times 10^8 \text{ m/sec} / \text{frequency}$ or $\lambda/4 = 75 / \text{frequency in MHz}$. Converting to inches, $\lambda = 2798 / f \text{ (MHz)}$. So, why are we using 2772 instead of 2798?

The speed at which a radio wave travels along a piece of wire is slower than in air or the vacuum of free space. That makes the wire act *electrically longer* than its physical length. In other words, $\lambda/4$ is shorter when the wave is traveling on a wire

than it is when the wave is traveling in free space. The thicker the wire, the slower the wave travels. This is called the *length-to-diameter (l/d) effect* and it must be accounted for when determining the length of antenna elements. For #14 wire, the l/d effect results in the use of 2772 instead of the free space value of 2798.

Cut five pieces of wire. Crimp a terminal onto one end of four of the pieces. Solder the remaining piece of wire to the BNC receptacle's center pin. Attach each radial to the receptacle with the #4 hardware.

Figure 4 shows one way to do it. Bend the radials down (away from the vertical element) about 45 degrees and arrange them symmetrically around the receptacle. Attach the coaxial cable to the antenna and the radio – you're done! **Figure 4** is a close-up of the antenna's *feed point* where the feed line is attached. (If



Learn at Home!

Electronics Courses

Cleveland Institute of Electronics

Train at home to be a professional electronics or computer technician! Learn with hands-on labs, instructor support, online exams, videos and instructor led chat rooms.

FREE course catalog www.cie-wc.edu or (800) 243-6446


• NEW! Robotics Automation Lab	• NEW! Computer Security Specialist
• Industrial Electronics with PLC	• Broadcast Engineering
• Electronics with PCC	• PC Troubleshooting
• Electronics Troubleshooting	• Networking

Visit www.cie-wc.edu and start today!

CIE: 1776 E. 17th St., Cleveland, OH 44114 • (800) 243-6446 • Registration 70-11-0002H

Create Your Own Custom Antenna!

UNIVERSAL ANTENNA HUB



Mount NMO
3/8-24
UHF
BNC
Patent Pending

THE ONE
& ONLY
**ALPHA-NODE®
HUB**

ALPHANODEHUB.COM

Ham Radio for Dummies, Second Edition (Ward Silver NØAX) — An introduction to ham radio that explains what it is and how it works in bite-sized chunks.

Ham Radio License Manual (ARRL) and *Technician Class FCC License Prep* (Gordon West) — The ARRL and other organizations publish study guides for all three exam licensing levels. Online study resources like www.hamradiolicenseexam.com and www.kb6nu.com/tech-manual are also available to help you pass the tests.

ARRL Handbook — Now in its 92nd edition, “the Handbook” covers nearly all areas of amateur radio technology from tutorials in basic electronics to the latest digital protocols and equipment innovations. See the ARRL Store (www.arrl.org/shop/technical) for a long list of technical books.

College students and educators will want to bookmark the web portal that supports university level ham radio interests at www.arrl.org/college-students-and-educators.

Ham
Radio
Resources

your radio uses some other type of connector than BNC, you’ll need to use an adapter.) If your radio has a signal strength meter, compare your new antenna to the flexible antenna provided with your radio. Why are the radial wires bent at an angle when **Figure 2** shows the ground plane as flat? If the radial wires (or conductive sheet) are at right angles to the vertical element, the *feed point impedance* of the ground plane will be about 35Ω which is different than that of the coaxial cable (which is usually 50Ω). This *mismatch* will make it harder for signals in the vertical element to transfer to the coax, and reduces the effectiveness of the antenna a bit.

By bending the radial wires down, the feed point impedance is raised closer to 50Ω . (Cable impedance will be the subject of a future column.)

Your antenna will perform well over a fairly wide range of frequencies that are up to $\pm 5\%$ from the design frequency. **Table 1** shows the wire length for several commonly monitored radio services. Assuming you don’t want to hold the antenna

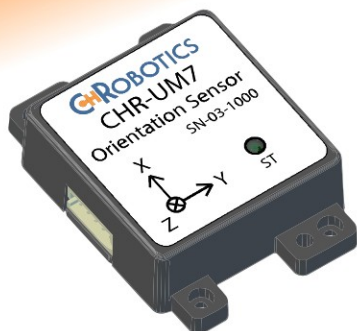
up in the air with one hand when you want to use it, you can simply tape the supporting cable to a piece of dowel, pipe, conduit, or whatever is handy. Secure the support so the antenna is in the clear and at least $\lambda/2$ away from any other metal surface or object.

I hope you’ve enjoyed this first installment of The Ham’s Wireless Workbench and will be a regular

reader as we explore the world of ham radio technology. Visit the ARRL website and read what’s there — you’ll learn a lot and maybe I’ll eventually hear you on the air with your own call sign! Until then, 73! (That’s the ham’s shorthand for “best regards.”) **NV**

Table 1 – Antenna Element Lengths		
Band Name	Frequency (MHz)	Length (inches)
Aircraft Voice	124	22.5
Amateur 2 meter	146	19.25
Public Safety VHF	157.5	17.8
Amateur 1-1/4 meter	223	12.6
Amateur 70 cm	446	6.3
Public Safety UHF	458	6.1
900 MHz ISM	915	3.1

◆ Inertial Sensors ◆ Orientation & Tilt Sensors ◆ GPS-Aided Attitude & Heading Systems ◆



Best-in-class performance
at
Unbeatable prices

Our new UM7 Orientation Sensor represents the state-of-the-art in low cost, light weight attitude and heading sensors. We’ve combined the latest and greatest MEMS technology with efficient in-house calibration processes, so that we can offer unrivaled performance without emptying your wallet.

Come and see how we are revolutionizing the inertial sensing industry!

www.CHRobotics.com

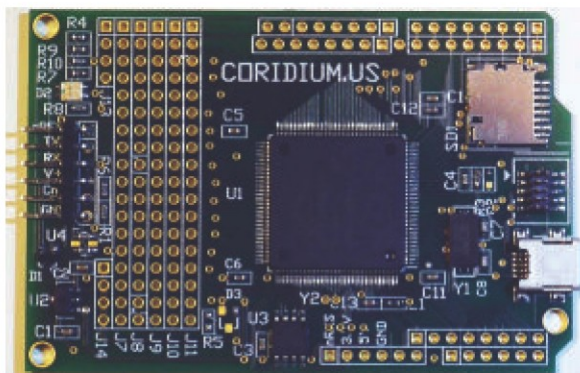
CHROBOTICS

www.CHRobotics.com

NEW

PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS



200 MHz DATA LOGGER

Coridium's newest data logger, priced at \$49, features:

- Dual ARM CPUs
- SD card Flash file system
- 83 inputs and outputs
- Four UARTs with DMA support
- Two I²S with DMA support
- Two CAN controllers
- Flexible GPIO interrupts

- Eight timers
- Motor PWM
- Quadrature encoder
- Two USBs
- Ethernet MAC
- Flexible serial GPIO capable of camera/display
- Optional RTC with battery backup

Technical specifications are as follows:

System

- 200 MHz ARM dual core M4/FPU M0+ CPU
- Floating point hardware accelerator
- Programmable in a compiled BASIC or C
- Operates faster than 50 million BASIC lines/second

Size

- 3.2" x 2.05"

Storage

- 264K RAM

- 8 MB Flash
- micro-SD card socket

Power

- 5V DC input connectors
- Arduino shield compatible
- Serial debug connector
- JTAG connector
- Connections for 83 digital I/O
- 13 x 12-bit analog A/Ds; one 10-bit DAC

Programming dongle sold separately

The data logger includes a 200 MHz dual core ARM with most every peripheral ever seen on one chip. It uses an NXP LPC4330 and brings out 83 I/Os and 13 A/D converters in an Arduino shield compatible form factor.

For more information, contact:

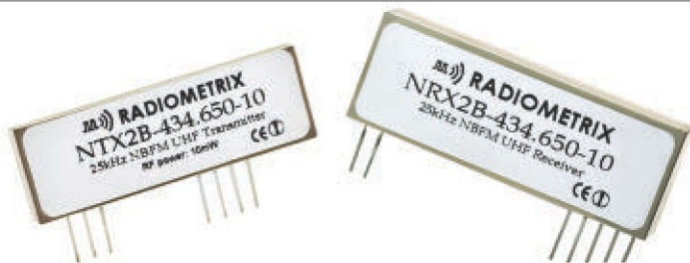
Coridium

Web: www.coridium.us

FREQUENCY PROGRAMMABLE NARROW BAND TRANSMITTER

Lemos International/Radiometrix has launched a new range of highly flexible/frequency programmable RF power adjustable radios. The new NTX2B transmitter offers industry-leading true narrow band FM performance and is available on user/factory programmable custom frequencies between 425 MHz and 470 MHz. Superseding the popular NTX2, the new transmitter offers greater stability and improved performance due to the VCTCXO reference. The NTX2B provides users with the ability to dynamically re-program the module via a microcontroller UART to other channel frequencies in the band, or store new frequency/power settings on EEPROM.

The standard NTX2B version is a 10 mW 25 kHz narrow band transmitter with data rates up to 10 kbps, and is available on 434.075 MHz, 434.650 MHz European SRD frequencies, and 25 mW on 458.700 MHz for the UK. The NTX2B is also available with 12.5 kHz or 20 kHz



channel spacing for licensed US FCC Part 90 or legacy European telemetry/telecommand bands.

The NTX2B features an internal LDO voltage regulator and can transmit both digital and 3V pk-pk analog signals.

Offering greater range than wideband modules, the transmitter can be paired with the new NRX2B receiver for a usable range of over 500 m — ideal for performance critical low power wireless applications including security, sensor networks, industrial/commercial telemetry, and remote control.

For more information, contact:

Lemos International

Web: www.lemosint.com

www.Primecell.com

Battery rebuilding service

Dead Batteries? Don't toss them. Send them to us - our rebuilds are better than original specifications.

Tools

Electronics

Radios

Visit www.primecell.com for important details.
 24 Hr Secure recorder tel-fax (814) 623 7000
 Quotes email: info@primecell.com
 Cunard Assoc., Inc. 9343 US RT 220 Bedford PA 15522

Recycling & Remarketing High Technology

WEIRDSTUFF®
WAREHOUSE

Software, Computers, Electronics, Equipment, Doohickies

WE BUY/SELL EXCESS & OBSOLETE INVENTORIES

FREE COMPUTER AND ELECTRONIC RECYCLING

GIANT 10,000 SQ. FT. AS-IS SECTION

**384 W. Caribbean Dr.
Sunnyvale, CA 94089**
 Mon-Sat: 9:30-6:00 Sun: 11:00-5:00
 (408)743-5650 Store x324
WWW.WEIRDSTUFF.COM

**1. Apply solder.
2. Listen.**

boxed
KIT AMPLIF

www.boxedkitamps.com

DIY audio kits
 stereo headphones guitar

In the Nuts & Volts Webstore NOW!

NUTS AND VOLTS

NV BOOK SPECIALS

Programming the Raspberry Pi
Getting Started with Python

Raspberry Pi
User Guide

Raspberry Pi
Projects for the EVIL GENIUS

Only \$48.95
 Plus FREE Priority Mail Shipping US Only

To order call 800 783-4624 or visit:
<http://store.nutsvolts.com>

Limited time offer

LE MOS
INTERNATIONAL

ZigBee Home Automation

ZigBee Pro

ZigBee Range Finder

www.lemosint.com

NATIONAL RF, INC.

Plug-in Coil Forms

WxSat Antenna

Analog Dials and Indicators

Visit www.NationalRF.com for this and other Radio Products!
 Office: 858-565-1319

ROBOT POWER™ Extreme Motor Speed Control!

MegaMoto Plus - Motor Control for Arduino™

- 6V-28V - 25A / 40A peak
- Single H-bridge or dual half
- Current and Temp protected
- Jumper select Enable & PWM
- Three units stack on one Uno

The VYPER™

- 8V-42V - 125A / 200A peak!!
- Single H-bridge
- RC and Pot control
- Overload protected
- For your BIG bots

www.robotpower.com
 Phone: 253-843-2504 • sales@robotpower.com

NKC electronics

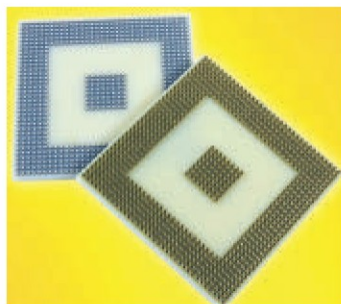
THE OFFICIAL ARDUINO STARTER KIT!

NKCElectronics.com/starterkit
 Includes the Arduino Project Book (170 pages)

Purchase Orders are accepted from Educational Institutions, US Government and Research Centers

BGA SOCKET ADAPTER FOR MOTHERBOARD

Ironwood Electronics has released a new BGA socket adapter for 724 pin BGA, with a 1 mm pitch, 35x35 mm IC size. The BGA socket adapter consists of two modules (female socket module and male adapter module). The patented female socket module was built with machined pins that match the male adapter module with similar construction. The SF-BGA724A-B-42F (female socket module) can be soldered to a motherboard using standard soldering methods without warping that results in reliable connection to a printed circuit board (PCB). Similarly, the



The NEW EZ HANG Square Shot Kit

www.ezhang.com

Suggestions from thousands of HAMs and Cable installers around the world led to a complete redesign of the EZ Hang. The New EZ Hang Square Shot will shoot a line over a 100' high tree. You can do all types of cable installations, and it will save you a lot of time and effort. It truly will be one of the most valuable tools for installing cable.

+ \$9.05 for shipping when paying by check
 Or order online with any major credit card and PayPal (U.S. Patent No. 6286495) • Call 540-286-0176

EZ HANG, Inc.

32 Princess Gillian Ct., Fredericksburg, VA 22406



A BARN DOOR TRACKER FOR ASTRO PHOTOGRAPHY

A barn door tracker — also known as a Haig or Scotch mount — is a device that slowly rotates a camera, allowing it to track the stars so it can take long exposure photographs of the night sky. High end astronomical telescopes usually have what's called a clock drive, which slowly rotates the telescope allowing for those long exposure photographs. The barn door tracker can take long exposure photographs with your camera without a telescope.

While it looks like the stars move across the night sky, it is actually the Earth's rotation that the barn door tracker must match. A barn door tracker gets its name from the two plates hinged together along one edge that when swinging, act like a barn door. One plate is attached to a tripod or other mount, while a motor driven threaded rod raises the other plate at approximately the same angular rate as the Earth turns. A small finder telescope is attached to help with aiming so that the axis of the hinge pin is pointed to Polaris, the pole star. This places the base plate at an angle from the horizon equal to the local latitude, with the hinge pin parallel to the Earth's axis.

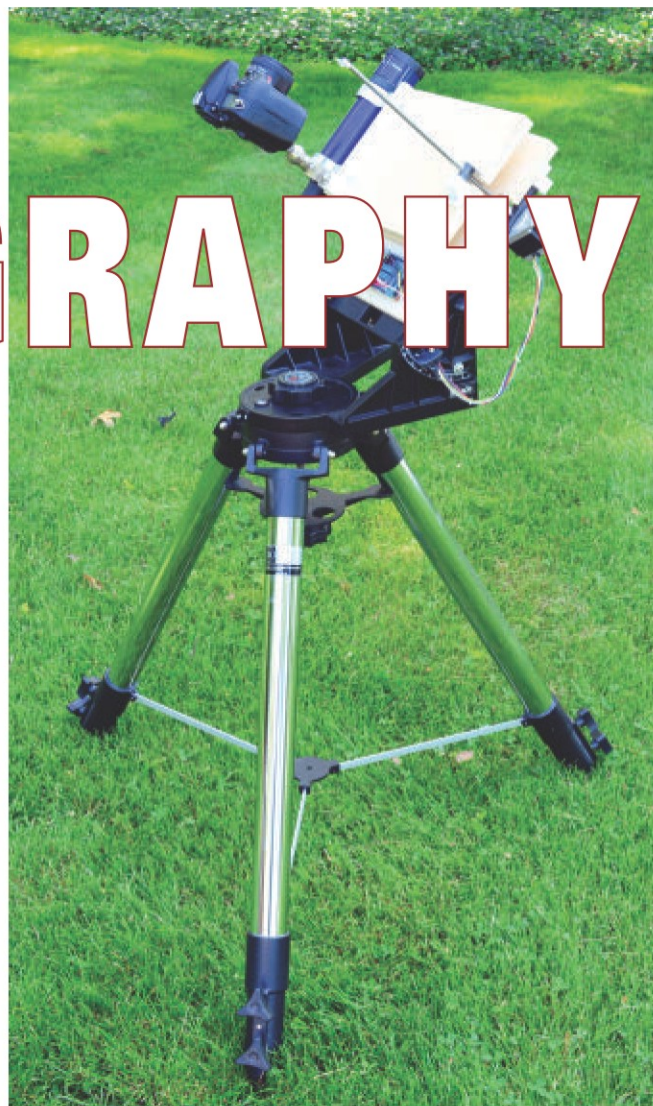


Figure 1 shows the basic geometry of the tracker. The threaded rod is attached to swivel mounts. (See the sidebar and construction section for more details.) Two pieces of cabinet grade 3/4 inch plywood are joined together by a hinge made from telescoping brass tubing. A 1/4-28 threaded rod joins the plywood plates. A stepper motor attached to the bottom of the threaded rod serves to turn it, which pushes up the top plate and rotates it about the hinge axis. A camera mounted on the top plate then rotates in synchronization with the Earth's rotation.

There is another detail we need to mention here. One would think the barn door should rotate at a speed of one rotation in 24 hours. Instead, it needs to rotate at the sidereal rate.

A sidereal day is 23 hours, 56 minutes, and 4.0916 seconds. As the Earth takes one solar day to rotate, it also is revolving around the Sun. Therefore, a star appearing directly overhead will take slightly less than one solar day to appear directly overhead again.

Construction

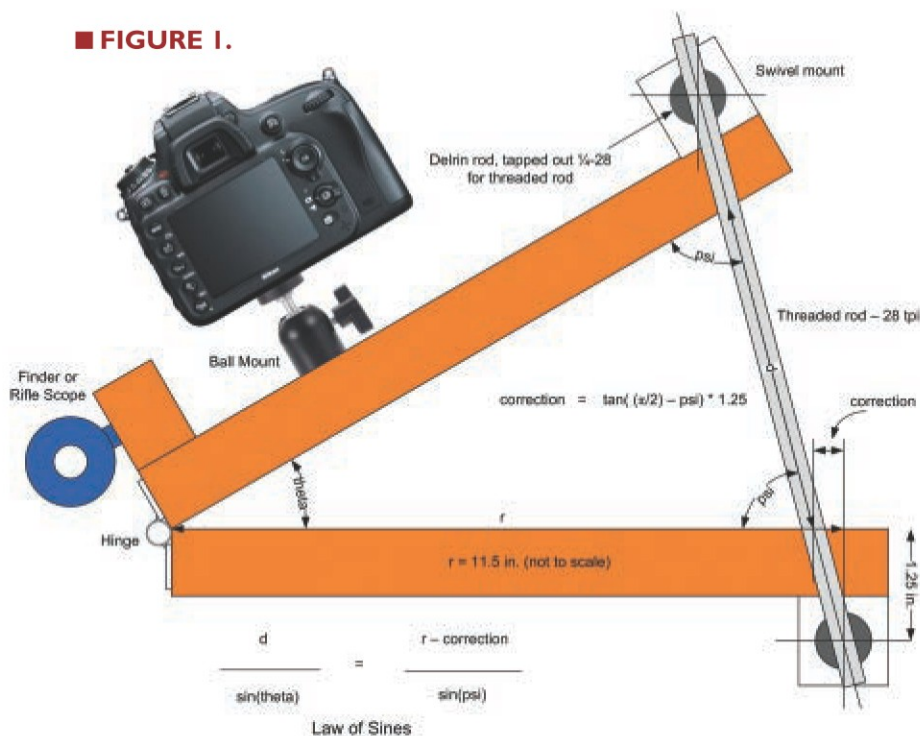
Referring to **Figure 2**, at the top the threaded rod goes through a pivoting nut made of 5/8 inch Delrin rod, although other plastics may work as well. A hole is drilled through the Delrin rod at a right angle to the rod's axis, and is then tapped out with 1/4-28 threads. The Delrin rod is mounted in L-shaped UHMW (ultra high molecular weight) polyethylene blocks that are screwed down to the top plate. This material is somewhat similar to Teflon but has better abrasion resistance.

At the underside of the bottom plate, the stepper motor is mounted to a longer piece of pivoting Delrin rod flattened on one side, and is mounted in L-shaped UHMW polyethylene blocks. A hole is cut through the Delrin rod for the stepper motor shaft. An aluminum coupling with two set screws attaches the threaded rod to the stepper motor shaft. As the top plate moves up, the pivoting nut must move to the left. This is accommodated by allowing both ends of the threaded rod to pivot, maintaining the geometry of the isosceles triangle.

There are many variations on how to increase the accuracy of a barn door tracker. This design computes the angle needed in real time several times a second to accurately move the camera at the sidereal rate. The rate at which the screw turns varies with the distance d , giving rise to difficulties in other designs. There are different approaches as to how to accommodate this varying rate.

The use of a piano hinge has been criticized and can lead to inaccuracies that show up in exposures as star trails. I used two pieces of brass telescoping tubing for the hinge with the outer tube soldered to brass strips, making a hinge that tracks more accurately. With the hinge pin being a small hollow tube, this is also a very useful feature in sighting the north celestial pole. **Figure 3** shows details on how the swivel mounts and hinge were constructed, and **Figure 4** shows the mounted improved hinge. By not cutting down the 12 inch hinge pin tube, the longer length works better for

■ **FIGURE 1.**



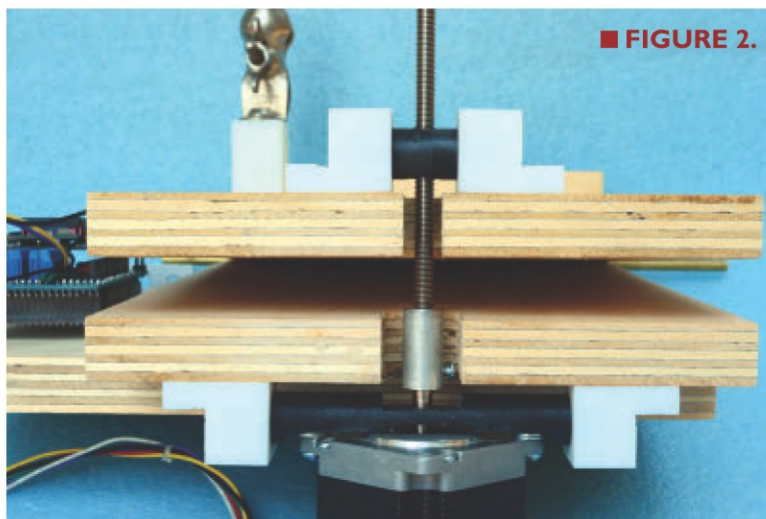
sighting Polaris. The bolts above the hinge mount a finder telescope.

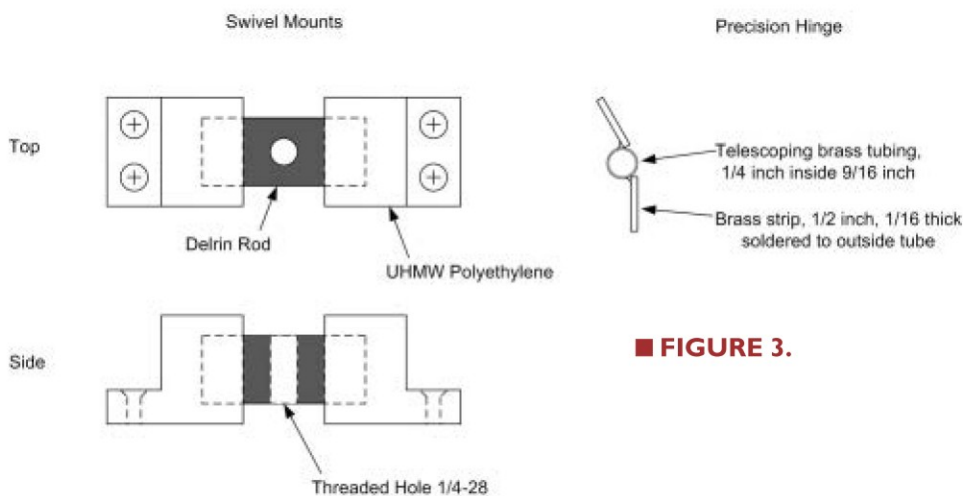
Figures 5 and 6 are views of the barn door tracker on a telescope tripod. Actually, any sturdy mount works as long as the hinge axis can be pointed at the north celestial pole, which is very close to Polaris. I used a finder telescope from an astronomical telescope, but something like an inexpensive air rifle scope could be used as well.

For the camera mount, I used a ball mount that screws into the 1/4-28 mounting hole (found on most cameras) which attaches to the top plate. This allows the camera to be pointed to the desired position in the sky.

It is important that the threaded rod is straight and

■ **FIGURE 2.**





■ FIGURE 3.

turns easily through the Delrin nut, and that both pieces of Delrin rotate easily in their UHMW polyethylene mounts. Any binding may cause the stepper motor to bind.

While it is possible to drive the stepper motor with an integrated circuit for that purpose, I choose to drive it directly with MOSFET transistors controlled from an Arduino board. The program will deal with the necessary signals to switch the MOSFET transistors on and off. The Arduino board and its associated shield are mounted

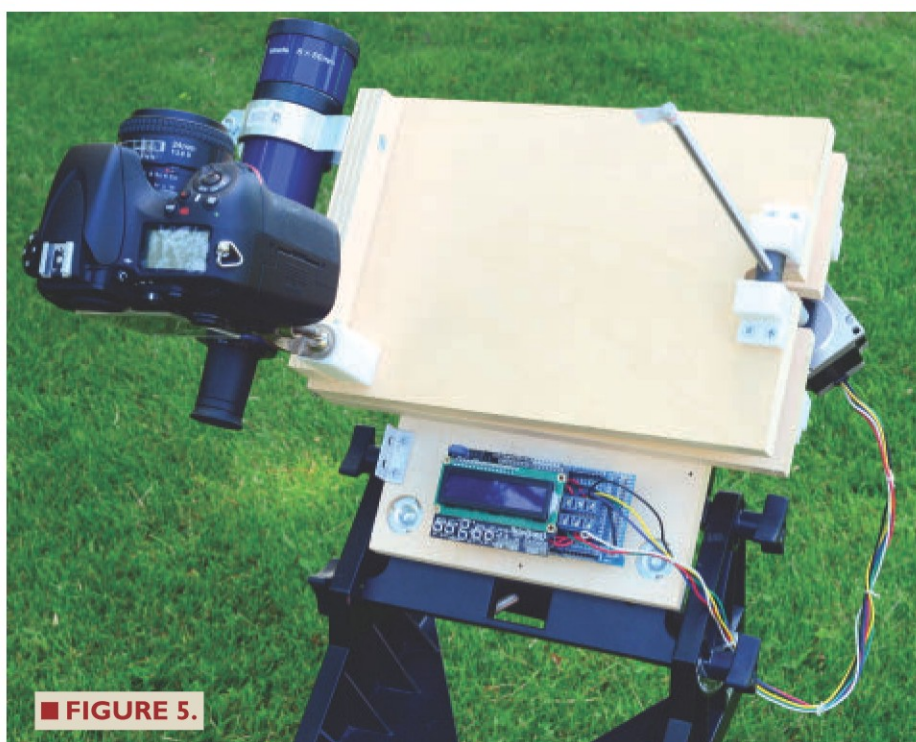
on the base of the barn door tracker. The Arduino board and additional circuitry can be powered with a 7.2 volt NiCd battery for portability or a standard plug-in power supply.

The Circuit

The schematic of the stepper motor driver is shown in **Figure 7**. The circuit consists of an Arduino board with four of its outputs driving MOSFETs that control the current



■ FIGURE 4.



■ FIGURE 5.

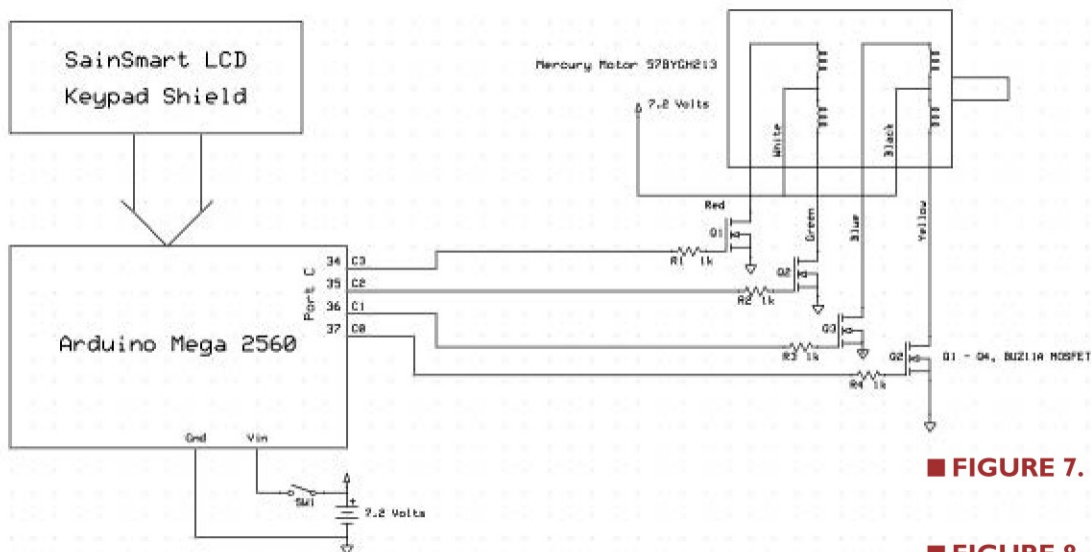


■ FIGURE 6.

through the four coils in the unipolar stepper motor. The MOSFETs are mounted on a Arduino Mega 2560 prototype board which is shown in **Figure 8**. The Mega 2560 board was chosen because the prototype board was larger and allowed for the easy positioning of the MOSFETs and the terminal posts. This allows the stepper motor wires to be easily

connected. The prototype shield used has a duplicate set of headers allowing for the insertion of a SainSmart LCD shield which is a convenient addition. We then have two piggyback shields plugged into the Arduino board.

The LCD shield allows for visual feedback on the progress of the stepper motor and its attached threaded rod. It also contains six small contact switches which are used to control the operation of the barn door tracker. The up and down switches set the direction of travel; Up being the direction used for tracking the stars at the sidereal rate. Down is used in conjunction with two other keys labeled Select and Left. The Select switch sets the stepper motor to track the stars, while the Left switch turns on a higher speed slew rate to return the barn door tracker to its starting position. These switches are connected to the A0 analog input port and are read using the `analogRead(0)` command. When tracking starts, the plates must begin in the closed position — the default position needed when the software starts running.



■ **FIGURE 7.**

■ **FIGURE 8.**

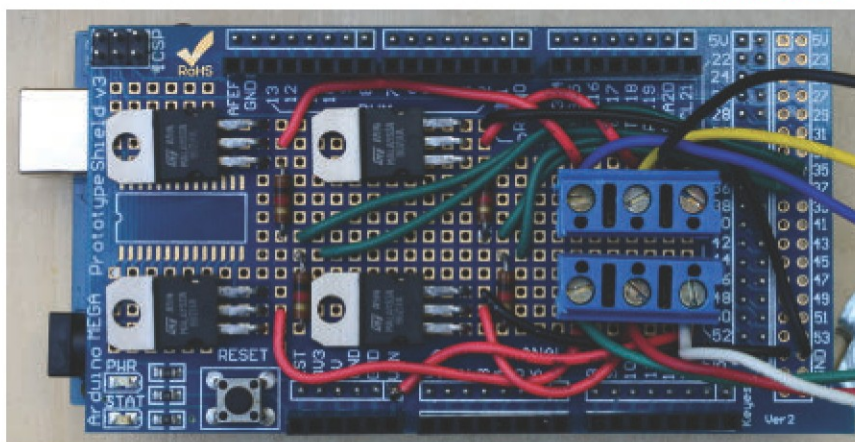


Figure 9 shows the SainSmart LCD keypad shield plugged into the prototype board. The output shown has the elapsed sidereal time on the first line, with the second line showing the number of steps taken and the angle of the two plates given in degrees.

A unipolar stepper motor was selected because it can operate from the same power source as the Arduino board, and the drive circuitry is quite simple. There are many surplus stepper motors available through electronics

ITEM

Mercury Motor 57BYGH213 or similar
 Arduino Mega 2560
 Arduino Mega 2560 prototype board
 SainSmart LCD keypad shield
 BUZ11A MOSFET transistors, 1K resistors, terminal posts
 Hookup wire, solder, 3/4 inch plywood, wood screws, machine screws, coupling between stepper motor and threaded rod
 Delrin rod, 5/8 inch, UHMW polyethylene
 1/4-28 stainless steel threaded rod, 12 inches long
 Telescoping brass tubing and brass strip
 Finder scope and mounts, or small rifle scope for an air gun
 Camera ball mount, tripod, or other base

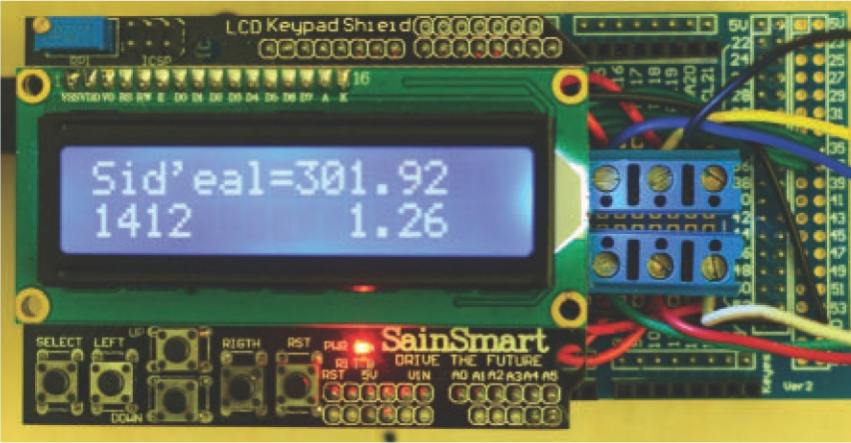
SOURCE

Electronic Goldmine
 Adafruit, Sparkfun Electronics
 Sparkfun Electronics
 SainSmart
 All Electronics

Home Depot, Lowes
 ePlastics
 eBay
 Tower Hobbies
 Amazon
 Amazon, B & H Photo

PARTS LIST

■ FIGURE 9.



dealers that advertise here and on the Internet. The only criteria when choosing a unipolar stepper motor is that the voltage be fairly close to the voltage used to power the Arduino board, and that it have enough torque to drive the threaded rod.

I selected the Mercury Motor 57BYGH213 because it is rated at one amp and 6.2 ohms, and has a 1.8 degree step. So, a 6.2 volt power supply will provide enough current. The torque rating is 9 kilogram centimeters which works very well. Voltages slightly higher than the 6.2 volts can be used without damage to the stepper motor. I use a nine volt plug-in power supply when not using the 7.2 volt NiCd battery. The BUZ11A MOSFETs chosen have internal diodes to protect them from the back inductive kick of the motor coils.

The Program

To get the stepper motor to turn, current needs to be

Step	Q4	Q3	Q2	Q1
1	Off	On	On	Off
2	On	Off	On	Off
3	On	Off	Off	On
4	Off	On	Off	On
1	Off	On	On	Off

Table 1.

Step	Q4	Q3	Q2	Q1
1	Off	On	Off	Off
2	Off	On	On	Off
3	Off	Off	On	Off
4	On	Off	On	Off
5	On	Off	Off	Off
6	On	Off	Off	On
7	Off	Off	Off	On
8	Off	On	Off	On
1	Off	On	Off	Off

Table 2.

sent through its four coils in a specific sequence. Table 1 shows the sequence for the steps that will turn the shaft clockwise when looking at the shaft. If instead, the order of the pattern is reversed, the shaft will turn counter-clockwise. Each step of the motor will turn the shaft 1.8 degrees, so 200 steps are needed to get a full revolution.

Table 2 lists a sequence of eight different patterns called the half step sequence. Sending this sequence will rotate the shaft one half of the step angle or 0.9 degrees, requiring 400 steps to turn the shaft one rotation. The following sets of data representing these patterns are stored in our

Arduino program:

```
byte fullsteps[4] = {0x06, 0x0A, 0x09, 0x05};
byte halfsteps[8] = {0x04, 0x06, 0x02, 0x0A,
0x08, 0x09, 0x01, 0x05};
```

Note that we are using only the low four-bit nibble of each byte. We will output each byte to port C, and connect the four wires going to the MOSFET gates to the lower four bits of port C. These are pins 34-37 on the Arduino Mega 2560; pin 37 is the least significant bit.

The program follows the algorithm shown in the sidebar. Using the geometry above, the program calculates how high the threaded rod must be raised given the current sidereal time, and then calculates how many steps of the stepper motor are needed. The program keeps a running total of the steps taken and then adds steps to raise the threaded rod the required distance. The complete program is available at the article link.

Accuracy

There are several things that will affect the accuracy of the barn door tracker. They are:

- 1. Most important is the accuracy in pointing the hinge pin at the north celestial pole.
- 2. The accurate measurement for the *r* value – the distance from the hinge to the center of the threaded rod.
- 3. The accurate measurement for the distance from the top of the bottom plate to the center of the bottom Delrin rod. This is used in the correction formula.
- 4. Step size of the stepping motor.

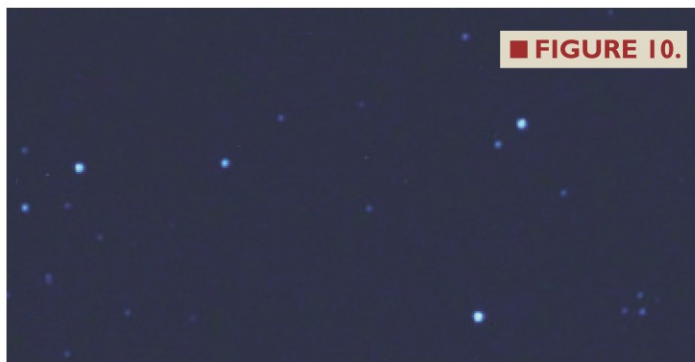
In reviewing this list, the accuracy of pointing the hinge pin and the physical measurements in 2 and 3 are going to have the greatest effect on accuracy. After building the barn door tracker, check these measurements. If there are slight differences, those corrected values should be used in the appropriate formulas. Some sort of finder scope attached in parallel to the hinge axis is also

helpful. Using a distant object in daylight or the moon, the finder scope can be adjusted to match the view through the hollow hinge pin. Viewing Polaris through the hinge pin can be difficult — especially with a bright sky in an urban environment. The step size can be changed to 400 steps per turn, but this improvement in accuracy is overshadowed by the first three items in the list. The dominant factor in accuracy is pointing the hinge pin at the north celestial pole.

The Results

Figure 10 is a portion of a four minute exposure at F/8 taken using the barn door tracker with a Nikon D810 digital camera, Nikon 50 mm F/1.4 lens, and a Lumicon high contrast filter to reduce the bright sky in my neighborhood. These shots are some of the first ones taken with the barn door tracker as a proof of concept, and are not the best quality. The stars shown are from the bend in the neck of the constellation Draco at approx. right ascension 19 hours, declination 70 degrees.

Figure 11 shows the same area with the barn door tracker turned off. Further details on taking photos with a barn door tracker can be found on Internet sites. **NV**



The Geometry

Referring back to **Figure 1**, look at the isosceles triangle created by the two wooden plates and threaded rod. The length of the base of the triangle created by the threaded rod is labeled d , while the length of the sides are labeled r . The angle created by the two wooden plates is labeled θ and the two equal base angles of the isosceles triangle are labeled ψ . Here is the sequence of the algorithm for calculating how high to raise the top plate at a distance d for a given elapsed time in solar seconds, beginning with the two plates closed creating an angle θ of zero:

1. Set $\text{solar_time} = 0$, $\text{total_steps} = 0$, $r = 11.5$.
2. Read elapsed solar_time in seconds.
3. Convert solar_time to sidereal time by multiplying solar seconds by 1.0027379.

```
sidereal = solar_time * 1.0027379
```

4. Calculate angle θ in radians. A circle contains $2 * \pi$ radians, or about 6.24 radians. Angles are calculated in radians because this is the unit necessary when using trigonometry functions in the Arduino program. Assume the barn door will run for at most three hours, or 10,800 seconds. Therefore:

```
theta = (sidereal_time / 10800.0) * (PI / 4.0)
```

where $(\text{sidereal_time} / 10800.0)$ is the fraction of three hours using seconds for both values, and $(\pi / 4.0)$ is the angle in radians θ that would subtend in three solar hours.

5. Next, calculate the angle ψ . The sum of a triangle's angles is 180 degrees, or π radians. Because the two base angles are equal in an isosceles triangle:

```
psi = (PI - theta) / 2.0
```

6. Using the Law of Sines which relates the lengths of the sides

of a triangle to the sines of its opposite angles:

$$d / \sin(\theta) = r / \sin(\psi)$$

$$d = r * \sin(\theta) / \sin(\psi)$$

7. The number of steps the stepper motor needs to have taken for this time is:

```
steps = (d * 28.0) * 200.0
```

given 28 threads per inch and 200 steps needed for each rotation of the threaded rod.

8. As the elapsed time progresses, keep track of the number of steps the stepper motor has completed, adding steps as necessary to maintain each newly calculated distance d :

```
while (total_steps < steps)
{
    have stepper motor do one step
    total_steps = total_steps + 1
}
```

9. Loop back to number 2.

The algorithm here works fine for the first few rotations of the stepper motor. However, as the threaded rod moves the top plate up, the rod itself starts to angle to the left on its swivel mounts. This changes the geometry as the distance r is reduced. The center of the swivel mount is located 1.25 inches below the top of the bottom plate. The angle of the triangle created when the threaded rod moves over is the complement of ψ , or 90 degrees ψ , which is $\pi / 2 - \psi$ in radians. These formulas are:

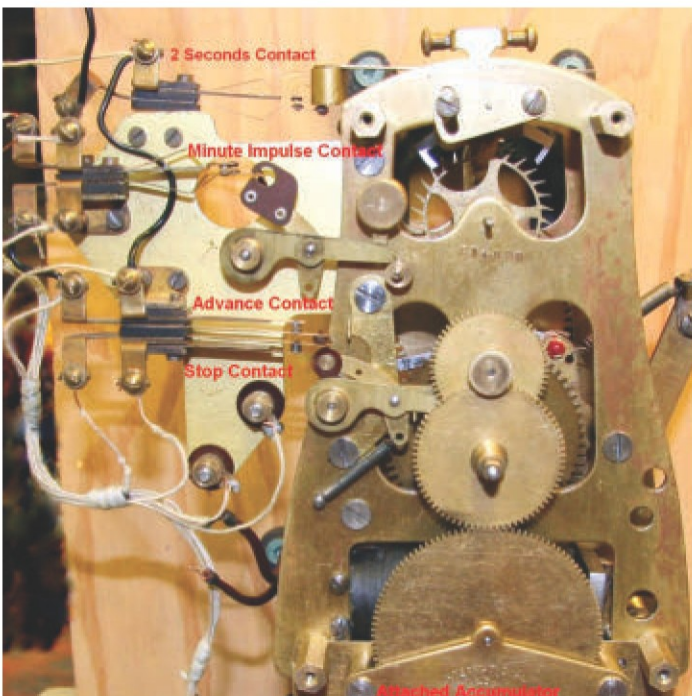
```
comp_psi = (PI / 2.0) - psi
correction = 1.25 * tan(comp_psi)
```

This correction amount must be subtracted from r before r is used in number 6 above.

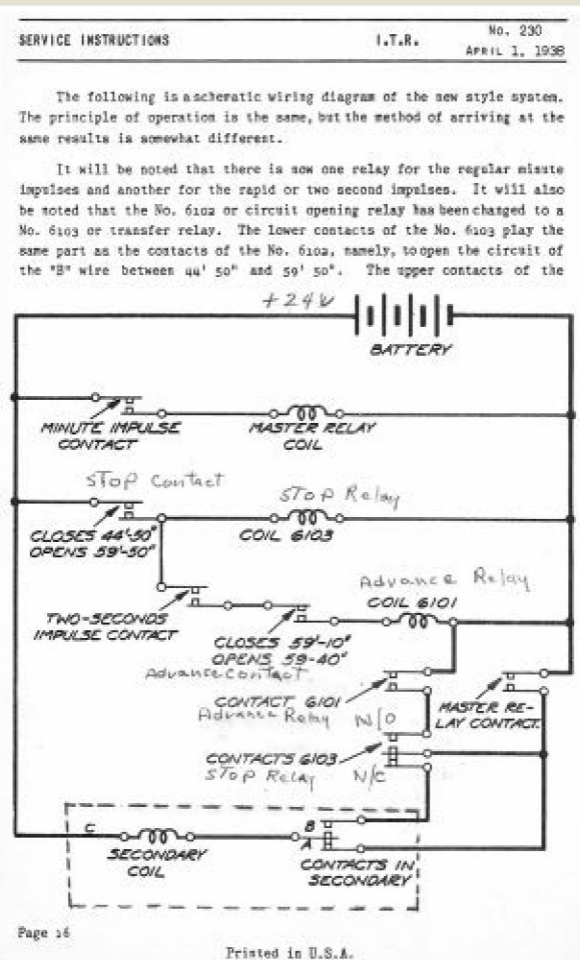
ARDUINO-BASED MASTER CLOCK



Over the years, I have seen many people on various forums asking if anyone has any information on how to get an old IBM slave clock working as a time piece. A master clock is required to control these old impulse secondary clocks (see **Photo 1**). These impulse secondary clock movements — along with time attendance recorders, bells, buzzers, etc. — were used for decades in many schools, factories, and businesses for coordinating time, so there are a lot of them out there. **Schematic 1** shows the basic wiring of an IBM time control system that was produced from about 1914 to December 1958 in the US.



■ **PHOTO 1.** IBM master movement, spring driven.



■ **SCHEMATIC 1.** I.T.R. Service Instructions, April 1, 1938, P16.

IBM sold its US Time Equipment Division to Simplex at the very end of 1958, but I still serviced these clock systems with IBM maintenance back in 1967. In 1966, I purchased the IBM master clock shown in **Photo 2** from a fellow IBMer, but my master had its contact assembly removed before I bought it. You can see original wiring and the contact assembly of a master I repaired for a friend in **Photo 1**. My master clock was originally installed at International Paper Company's Southern Kraft mill in Panama City, FL in 1930, but because of the missing contact assembly it can no longer control secondaries. Two of my most cherished secondaries (**Photos 3** through **6**) came from the same paper mill, and another one (shown in **Photo 7**) came

from Cove Elementary School where I attended the second through sixth grades.

Since the day I acquired my first IBM secondary clock movement, I've wanted my own clock system. I did manage a couple of feeble attempts at making my own contacts, and even purchased a BASIC Stamp for the project. However, I procrastinated (about 46 years) before finally building this computerized version of a master. Of course, the Arduino Mega 2560 I received as a Christmas gift helped motivate me.

These electronic masters (**Photos 8** and **9**) are far better and much more accurate than the brass, nickle, and mercury master clock living in my foyer. Even better, I now have all five of my IBM secondaries clicking away in my

This article describes a computerized equivalent master clock for an IBM minute impulse self-regulating clock system. So, if you are lacking a functional master, build this project and get one or more IBM impulse secondary clocks working accurately and electronically. Information here may also help you control other items by using the popular Arduino project computer and an insulated-gate bipolar transistor (IGBT) MOSFET device. For instance, you could drive a high current unipolar stepper motor winding with the Arduino and an IGBT device the same way.



■ **PHOTO 2.** My 1930 IBM master clock.

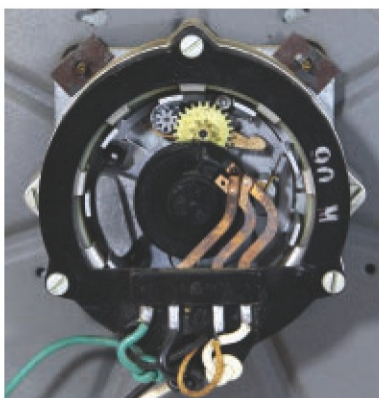


■ **PHOTO 3.** Impulse secondary from executive offices. Soft copper shell, brass numerals, scripted "International," and minute markers each screwed to a nickle face.



■ **PHOTO 4.** Impulse secondary from engineering dept.; soft copper shell and painted face.

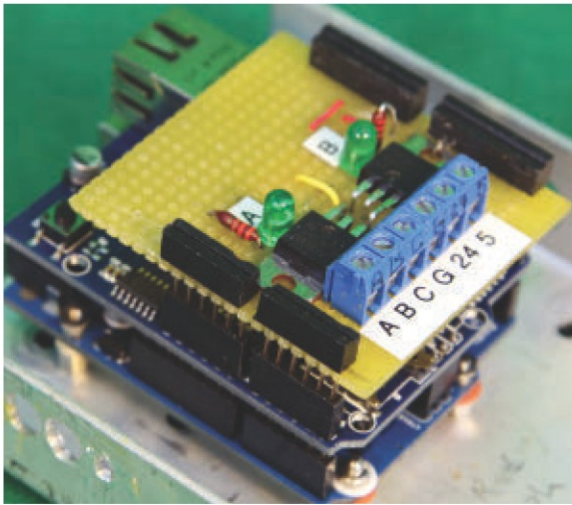
■ **PHOTO 5.** Model 561-2 movement of clocks in Photos 3 and 4.



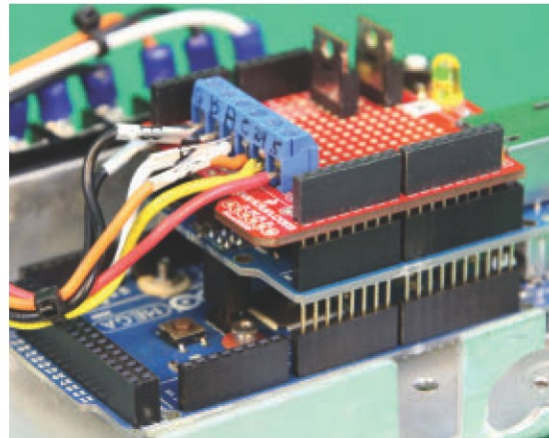
■ **PHOTO 6.** Model 563-2 movement of clock in Photo 7.



■ **PHOTO 7.** Impulse secondary from Cove Elementary.



■ PHOTO 8.Arduino UNO with fabricated driver shield.



■ PHOTO 9.Arduino Mega 2650 with SparkFun protoboard.

garage, all displaying the exact same correct time.

You can expect to spend around \$35-\$75 or more to build this project depending on your junk box, building your own shield and power supply, or where you buy your Arduino. The impulse secondary driver circuit shown in **Schematic 2** uses the IRFZ44N – an IGBT device. This transistor is perfect here and for other control projects because the gates of these devices can be connected directly to the Arduino's output pin and driven with 4.5 to 10 volt logic levels. The LEDs and their resistors are actually optional, but they let you know if and when an Arduino's output is active.

Either the Arduino UNO or Mega 2560 model can be used for a master clock, along with an Ethernet shield. Both Arduino and SainSmart Ethernet shields are

completely compatible and interchangeable – even after the Arduino is programmed. This configuration allows the use of the Network Time Protocol (NTP) to update the Arduino's system timer (system clock) once each hour.

Neither of the Arduino's system clocks are accurate

enough to be a clock time base (not even close).

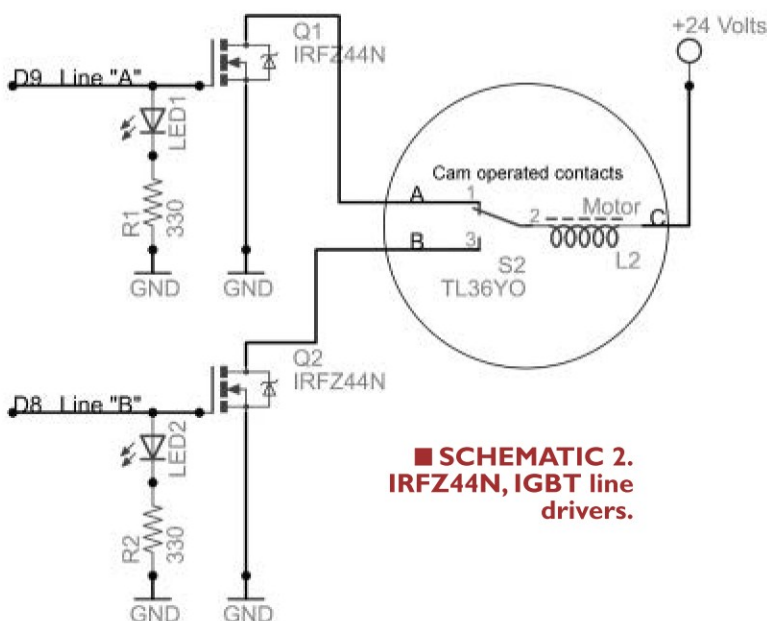
The IBM master clock with an Invar steel pendulum and compensating mercury pendulum weight (like mine) had a guaranteed accuracy of 10 seconds per month back in the 1930s. Because the Arduino's system timers gain or lose about a second per hour with current code (around 12 minutes per month), an external time base source is required. In fact, my first attempt on the UNO was 17 seconds slow per hour. So, after looking at GPS, temperature controlled crystal oscillators, real time clocks (RTC), WWV (radio) modules, and an Internet Time Server as possible sources, I chose the cheapest and most maintenance free over time: the time server.

The Ethernet shield accesses the Wide Area Network (WAN) with an NTP time source request from the official US government's time keepers: the National Institutes of Standards and Technology's (NIST) Internet time server.

This is an accurate and reliable clock time base choice for the Arduino master. The Arduino's system timer (clock) is still used, but its timer is corrected every hour by the NTP time request two minutes before synchronizing all the clock movements at the top of each hour.

In other words, the master clock will never be off more than about a half second, regardless of temperature, voltage, crystal aging, and other variations. Bear in mind the impulse secondary has a resolution of one minute, not one second. Only the master clock has a resolution of one second. The impulse secondary will only be on time for one second, then will become up to 59 seconds slow until it is brought forward at the next impulse, at zero seconds of the next minute.

There is useful information at <http://tf.nist.gov/tf-cgi/servers.cgi>. Be sure to heed their warning that: "All users should ensure that their software NEVER queries a server more frequently than once every four seconds. Systems that



■ SCHEMATIC 2.
IRFZ44N, IGBT line
drivers.

exceed this rate will be refused service ... "

The Arduino-based version of the IBM minute impulse self-regulating clock system works by outputting two signals. Starting at the top of the hour, an A and B pulse are presented for a duration of 0.4 seconds on, then off once each minute, with the following exception. The A signal is raised once per minute at zero seconds, and on every odd second between 10 and 50 during the 59th minute. Secondaries that are fast (due to mechanical failure) will receive the A pulse once per minute until their 59th minute. Then, they switch from their N/C (A) contact to their N/O (B) contact, and will only advance to the top of the hour on the next B signal.

This secondary will switch back to the A contact between four to 14 minutes after the hour, depending on the model of the secondary. The B signal is raised once per minute at zero seconds for each minute, except for minutes 50 to 59. Secondaries that are late (due to mechanical failure) will receive the A pulse once per minute until the system is at the 59th minute, then receive up to 21 rapid A pulses every other second until the late secondary reaches its 59th minute. The secondary then switches its contacts to the B contact and waits for the next B pulse.

The circuit I use to drive the clocks is very simple and uses an IGBT transistor to apply a ground to the secondary's coil. I connect one side of the secondary's coil through line C to +24 volts (for 24 volt coils) and take lines A and B to ground to operate the electromagnets. This is the same logic as the wiring of the relay technology wiring design shown in the 1938 IBM diagram in **Schematic 1**. This allows me to use a common and cheap

high current IGBT transistor whose gate is turned on by the Arduino's output pin (refer again to **Schematic 2**). Perusing the IRFZ44N datasheet from **jameco.com**, you will find the Source to Drain Current vs. Gate to Source Voltage graph in the datasheet's Figure 3. Notice that 4.5 volts at the gate should turn on the IGBT to about nine amps — enough to run about 375 model 561-2 secondary movements (**Photo 4**). This is more than the two amp limit of IBM's electrical relay system could handle.

The IGBT source terminals are connected to the ground terminal (the -24 volt and -5 volt power supply return), and the drain terminals are connected to the line A and line B terminals. I placed a jumper from the +24 volts terminal to the line C terminal on the driver shield, and included the LEDs and their limiting resistors at each IGBT gate to show me the state of the lines. You will see the A and B LEDs light each minute, the rapid A pulses beginning at the 59th minute:10 seconds, and the B LED not illuminate between the 50-59th minute. The impulse secondaries are all wired in parallel to complete the system wiring: A to A, B to B, and C to C.

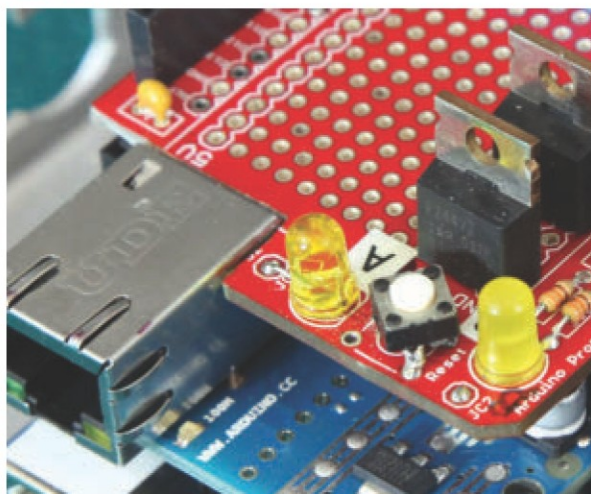
The Arduino's maximum sourced current per output is 20 mA. Maximum total current for all outputs combined is 150 mA, or 7-1/2 outputs. I believe the IGBT's gate current load presented to an Arduino I/O pin by itself is so small that it is negligible. It is the electromagnetic field that turns the transistor on, and practically all of the I/O pin's load in this circuit is from the LED-resistor pair. Go to **<http://playground.arduino.cc/Main/ArduinoPinCurrentLimitations>** for more information. The SparkFun prototyping shield schematic shows a 330 ohm resistor that limits the LED current to about 15 mA.

ITEM	PART#	COST**	SOURCE	SOURCE
RS405L	P/N 1541466	\$0.89 ea	Bridge Rectifier, 1,000V PIV, 600V @ 4A	www.jameco.com/Jameco/Products/ProdDS/1541466.pdf
L7805CV	P/N 924633	\$0.45 ea	5V Regulator, 1.5A	www.jameco.com/Jameco/Products/ProdDS/924633.pdf
LM7824T	P/N 51414	\$0.35 ea	24V Regulator, 2.2A	www.jameco.com/Jameco/Products/ProdDS/51414.pdf
IRFZ44N	P/N 669951	\$1.25 ea	N Channel Power MOSFET, 55V, 49A, TO-220AB	www.jameco.com/Jameco/Products/ProdDS/669951IR.pdf
OSTTA024163	P/N 152347	\$0.69 ea	Term Block 2 Pos, 5.08 mm Solder Thru-Hole 15A	www.jameco.com/Jameco/Products/ProdDS/152347.pdf
LED <i>see note</i>	P/N 1554161	\$3.80 20pk	Red LED, low current, 7 ma forward current. LED and resistor optional. Use 680 ohm resistor to limit max current to 7ma.	www.jameco.com/Jameco/Products/ProdDS/790241.pdf
Resistor, LED	P/N 690822	\$0.99 10pk	680 ohm resistor to limit LED current to 7 ma	www.jameco.com/Jameco/catalogs/c142/P39.pdf
Perfboard	P/N 616690	\$5.95 ea	6" x 4.5", 0.1" pitch, 0.042" dia holes	www.jameco.com/Jameco/Products/ProdDS/616690.pdf
Arduino Ether	P/N 2152375	\$37.95	Arduino Ethernet shield w/o PoE*	www.jameco.com/Jameco/Products/ProdDS/2152374.pdf
From SainSmart.com : SainSmart EthP/N W5100 (\$19.99)/SainSmart Ethernet shield for UNO and Mega 2560				
From SparkFun.com : Proto shield P/N Dev-07914 \$9.95 ea				
Modification required to plug into Ethernet shield; refer to Photo 13 . www.sparkfun.com/tutorials/191 www.sparkfun.com/datasheets/DevTools/Arduino/ProtoShield-v25.pdf				
			*Power over Ethernet ** Prices subject to change without notice.	

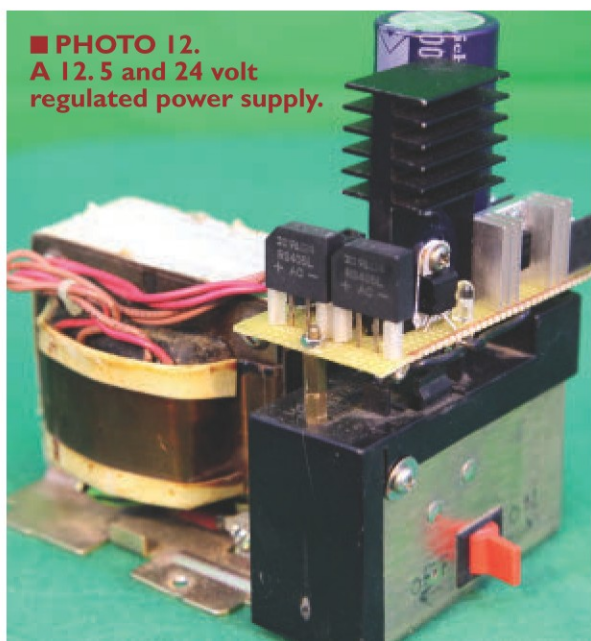
**PARTS
LIST**



■ **PHOTO 10.** Cutting wire wrap socket with hacksaw.



■ **PHOTO 11.** Close-up of SparkFun's shield. Modification done to clear the Ethernet connector.



■ **PHOTO 12.**
A 12.5 and 24 volt
regulated power supply.

This seems to me to load the Arduino's output pins a lot, but I am only using two outputs here. Be aware, you should consider using low current LEDs with 680 ohm resistors to reduce the output pin loading to about 7 mA per output. Although the schematic lists 330 ohm resistors, I have listed low current LEDs and higher values for the series resistors in the **Parts List** to reduce loading and maybe reduce chip heating.

You can purchase a prototyping shield for about \$10 and up, but I make my own. Simply use a piece of perfboard (0.100 inch [2.54 mm] holes center to center) the size you need, and two or more 16-pin dual inline wire wrap sockets. As a true hacker, I save time and shipping costs, which is important when you are on a fixed income.

A nice advantage is that you can make the board as large as you want, and add additional socket pins for headers as needed. I just saw (cut) the wire wrap sockets in half with a hacksaw blade, and glue the halves to the board with a very small dab of epoxy glue between the board and socket plastic. Wait until the glue is fully cured, then connect everything with 26-28 gauge or smaller hookup wire to the wire wrap pins on the back side. The wire wrap pins are long enough to give the perfboard clearance to everything on the Ethernet shield and leave room to either wire wrap or solder wires above the Arduino's or Ethernet shield's headers.

I did have a little bit of a problem with the SparkFun prototyping board and had to modify it by cutting off the corner along with a resistor and the S1 button switch to get the needed clearance for the Ethernet connector. Using the wire wrap socket pins, my board cleared the Ethernet connector entirely.

For the power supply, you can use the nine volt DC input jack to the Arduino's regulator to power the Arduino, or use five and 24 volt DC wall warts connected to the driver shield terminals. I made my own "brute force" power supply (**Photo 12**) by using an IBM Proprinter transformer, having a 28.1 volt and 8.8 volt secondary winding as shown in **Schematic 3**. If you can find a suitable transformer, the two bridge rectifiers (\$0.89 ea), a 24 volt regulator (\$0.45 ea) and a five volt regulator (\$0.35 ea), along with hacked electrolytic capacitors large enough to maintain the correct output voltages, that is an inexpensive alternative. Heatsinks are recommended on these regulators because they can get hot, depending on input voltage. To transistors, heat is an enemy. Some will question why I don't use more efficient switching power supplies. In my case, \$2.53 vs. \$40 was the reason.

Software

The code works on both the Arduino UNO and Mega 2560 model computer boards, and the Arduino and SainSmart Ethernet shields. Both Ethernet shields have access to the same Arduino I/O pinouts (D9 and D8), so my shields were wired to digital output pins common to both Arduino models that did not interfere with the Ethernet shield's communications. Note that the Ethernet shields use digital pins D10, D4, D3, D2, D1, and D0 to communicate with the Arduino computers.

Luckily, I have a software magician in the family — my son-in-law, Phil Hord — along with two grandsons following in his

Master Clock code written by Phil Hord; Phil@phord.com. Code not obtained from the Web or other sources is granted to public domain, December 25, 2013.

Arduino is a registered trademark of ARDUINO, LLC, Cambridge, MA 02141

IBM, I.T.R., are registered trademarks of International

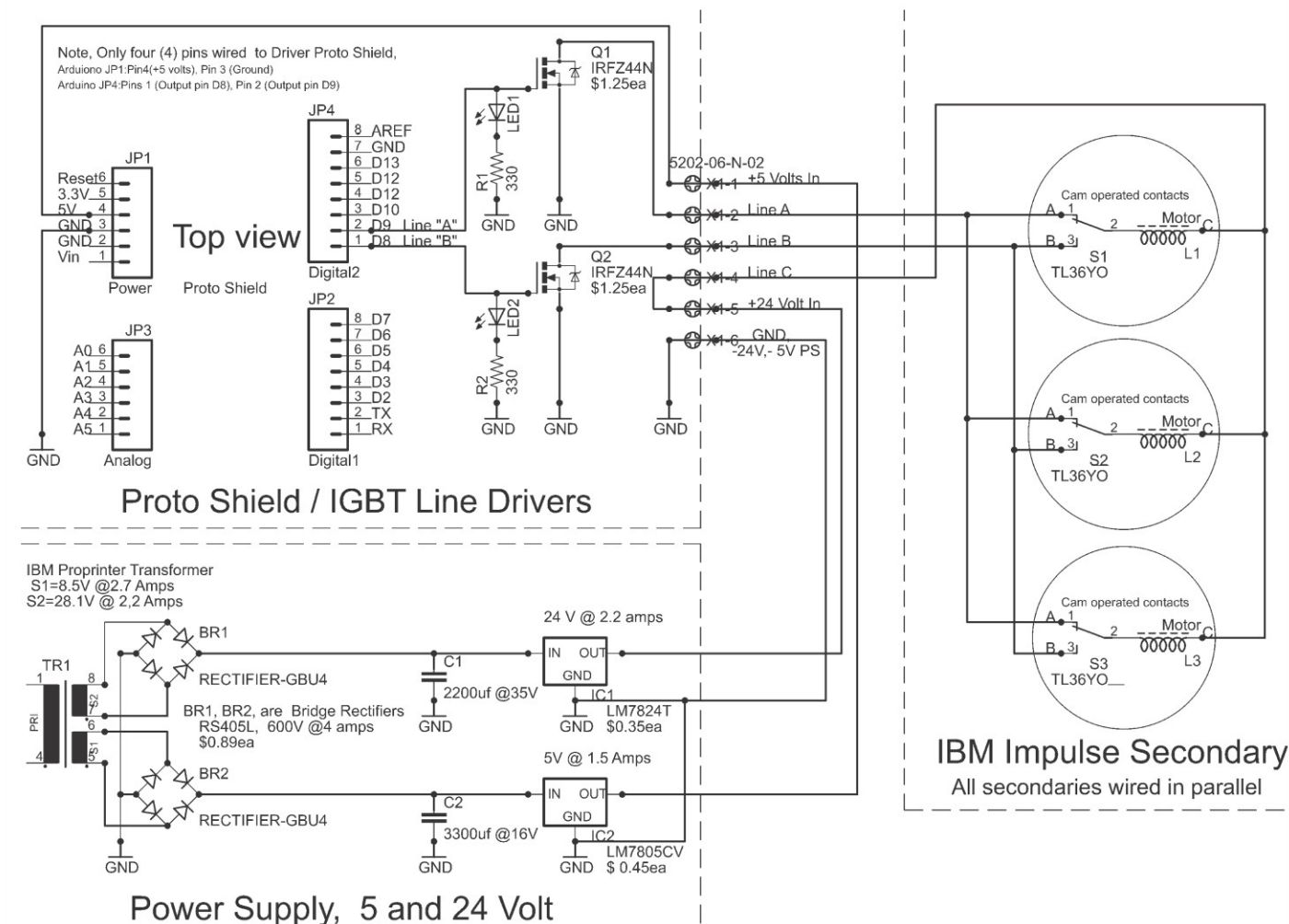
Business Machines, Armonk, NY

Page 16, I.T.R. Service Instructions No 230, April 1, 1938, re-printed with permission from:

Reference Desk
IBM Corporate Archives
Route 100/CSB
Somers, NY 10589
914-766-0612
archive1@us.ibm.com

initializes software variables, performs several tasks, then settles into a loop. Provisions are made to control the program from the keyboard via the serial monitor — if the Arduino is plugged into the USB port and the serial monitor is activated. This master clock will run completely by itself if it has an Internet connection with a DHCP server, is connected as in **Schematic 3**, and is powered on. If you are using the serial monitor (referring to **Photo 13**), the screen will show the minute:seconds displayed every second, and will scroll up the page. You can uncheck auto-line feed at the bottom of the screen if you want to look at previously scrolled pages of the display.

At 58 minutes, zero seconds, the A line active event, the NTP request, the programmed MAC address, the NTP 48 byte packet received, system clock correction, and the line "off" event will be displayed. Then, a display of each minute:second as the loop runs is shown. Typing on the console will enter the commands as mentioned in the console code. My cropped screenshot shows both the UNO and Mega master clocks running at the same time.



■ **SCHEMATIC 3.** Arduino-based master clock system: driver shield, power supply, and clock wiring.

■ **PHOTO 13. Screenshot of two serial monitors showing both Arduinos and Window of NIST official time web page.**



Notice each has a unique MAC address.

You will need to install the timerone.zip library using the instructions from the Arduino group web page at <http://arduino.cc/en/Guide/Libraries>. Get Timerone from <https://code.google.com/p/arduino-timerone/downloads/list>. Important: Rename the current revision of timerone-r11.zip to timerone.zip, or it will not load into the IDE's library folder. The IDE does not like the "-r11" part of the name. Then, compile the Arduino sketch and upload the generated code into your Arduino. The following is Phil's description as to how some of his code works. According to Phil, here is how the files are laid out:

master_clock.ino

This is the main Arduino code. It defines special functions that work only on the Arduino.

clock_generic.ino

The main clock protocol implementation. It advances the time and performs functions based on the minutes and seconds counters.

console.ino

The console code. This code talks to the serial port to show you the time and activity output. It also reads the keyboard input to allow you to advance the time, set the time, change the clock speed, and so on.

Udp.ino

A simple Udp "abstraction" for the Arduino. This code is only used on the Arduino.

ntp.ino

The NTP code. This is separated from the UDP code so it can be used and tested on Phil's PC.

The following files are all "header" files for the real code. They define prototypes for the actual functions used later on. Doing this helps the compiler ensure that all the types are correct in the end.

```
clock_generic.h
console.h
myUdp.h
ntp.h
```

The NTP time request is set up to occur at 58 minutes after the hour. This way, if the Arduino clock is running fast or slow, it will be corrected near the top of the hour, and the updated time will be reflected on the real clock using the A-B signal lines protocol in just a couple of minutes. If there is any inaccuracy in the timer, it should not accumulate much error in only two minutes.

You will be interested to see how some of this works. Have a look at generic_clock.ino. At the end of this file is the "service" routine. This is like the Arduino loop function that gets called over and over, forever and ever. At the start of this function, you will see this:

```
// the service routine runs over and over again
// forever:
void service() {
  consoleService();
  checkNtp();

  switch (state) {
  default:
  case rise:
    a = checkA();
    b = checkB();
```

The first two lines of the function call out to the `consoleService()` and `checkNtp()` functions. These functions are therefore also called over and over, forever and ever. Now, the interesting part is in `checkNtp`. This function is contained in the same file:

```
void checkNtp() {
  static bool needResponse = false;
  static bool sentRequest = false;

  if ( m == 58 && !sentRequest ) {
    // Read the NTP server once per
    // hour
    sendNtpRequest();
    needResponse = true;
    sentRequest = true;
  }

  // Get ready to try again on the
  // next hour
  if ( m == 2 ) sentRequest = false;

  if ( needResponse ) {
    int mm, ss;
    bool success = readNtpResponse
      ( &mm, &ss );
    if ( success ) {
      p("\nNTP Server: adjusting time
        by %d seconds.\n",
        (mm*60 + ss) - (m*60+s) );
      setMinutes(mm);
      setSeconds(ss);
      needResponse = false;
    }
  }
}
```

Console

The serial monitor is used when testing a secondary for problems, etc. You can send any of these commands from the serial monitor. On the serial console with the

Program code libraries were obtained from the Internet.

1. Ethernet Shield Sketch came from <http://arduino.cc/en/Tutorial/UdpNtpClient>. Note: If you have two or more Arduino (SainSmart, etc.) Ethernet boards on your LAN, change the MAC address of one of them [assigned in the *UdpNtpClient* code] before compiling and uploading to the Arduino, so each will have a unique MAC address. See:

```
// Enter a MAC address for your controller
// below.
// Newer Ethernet shields have a MAC address
// printed on a sticker on the shield
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

Arduino IDE, you have to press ENTER to send text, but the ENTER does not get sent itself. So, after entering a new time value, you have to enter something else (like a “,” or “.” or anything) to get the time to set. The one-character commands just need the ENTER since they work in just one byte.

The Advance/Retard and Force-Pulse commands can be buffered. So, you can type "+++++ ENTER" to advance five minutes, or you can type "ABABAAAA" to send two AB pulses and four A pulses.

+ Advances the minute counter (internal only)
- Decrements the minute counter (internal only)
Z Zeroes the seconds
A Forces an A pulse

B Forces a B pulse
C Forces both A and B at the same time (works just like 'AB')

12:34; Set the minutes and seconds to 12 and 34

1234; Set the minutes and seconds to 12 and 34

12;; Set just the minutes to 12; leave seconds alone

:34; Set the seconds to 34; leave the minutes alone

Changing the seconds (with Z or :34;) also syncs the clock to align with the second start. I added a few new console features while testing this out. If you press the left and right angle bracket keys, it will make the clock speed go up or slow down by 2x each time. This is not for running time correction; it is only for testing.

Press ">" to speed the clock up twice as fast.

Press "<" to slow the clock down half as fast.

Press "N" to trigger the NTP clock synchronization manually " (close quote, program notes)

I hope you'll enjoy working with secondary clock movement as much as I do. Now, go have the time of your life. **NV**



4Discovery

3.5" Wall/Panel mountable Intelligent Display Controller

- 3.5" TFT LCD Display
- Resistive Touch
- Wall and Panel Mountable
- Powerful Diablo16 Processor
- RS485 Interface
- Optional WIFI

NEW



Description

The 4Discovery is a high resolution 480x320 pixel 3.5" Intelligent wall mounted display product from 4D Systems, featuring the DIABLO16 Processor.

The 4Discovery is designed to be mounted to a standard light switch flush/mounting box, which enables quick and easy installation in to a wall, and can just as easily be mounted into an enclosure or panel, enabling a wide range of customers to take advantage of this display product.

The 4Discovery has an array of features including micro-SD memory storage, 16MB of Flash Memory storage, Real Time Clock, 2 wire RS485 Interface which can act as Master or Slave with additional changeover wire, Optional WIFI, Optional Crypto Authentication security chip for secure transmissions, and a switch-mode power supply enabling a wide input voltage range, along with many more features.

Dimensions

Viewing Area	48.96 x 73.44 mm
Module Width	74 mm
Module Height	117 mm
Module Depth	22 mm
Module Weight	80 g

Available As:

- 4Discovery-35 (Non WIFI)
- 4Discovery-35W (WIFI enabled with Crypto Authentication)

www.4dsystems.com.au

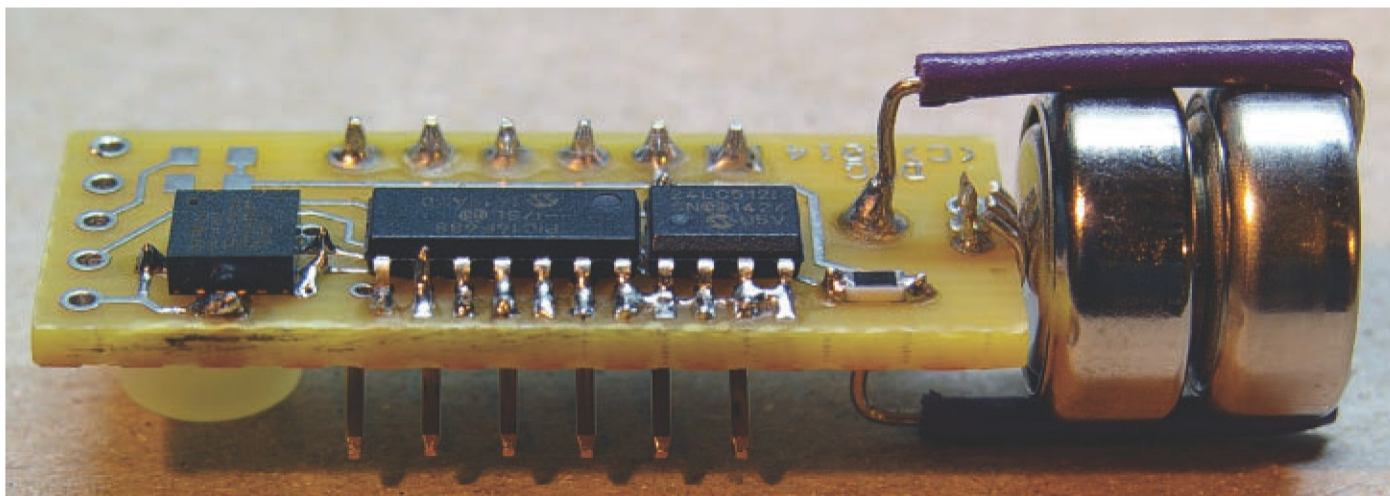
[.com/4DSystemsAU](https://www.facebook.com/4DSystemsAU)

[.com/4DSystems](https://twitter.com/4DSystems)

FLIGHT COMPUTERS FOR SOUNDING ROCKETS

By Dan Gravatt

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?magazine/article/january2015_Gravatt.



Launching model rockets is a lot of fun for kids and adults alike. Kids love the fire and smoke and trying to find the rocket once it's landed. The bigger they are and the higher they go, the better. So, how high did it go, exactly, and what did it do along the way? What happens when the rocket drifts to Earth out of sight? The flight computers described here have the ability to collect data; control other payload instruments (cameras, sensors, radios, etc.); trigger activities in flight, such as igniting a second-stage motor or deploying parachutes or glider wings; send simple telemetry data via a built-in transmitter; and act as a radio beacon to find the rocket once it's landed. With one of these as your payload, you're not just entertaining the kids — you're launching a sounding rocket and doing real science.

There are four versions of this flight computer project with varying sizes and complexity, but each has the same core functionality: a PIC16F688 microcontroller, a 64 KB EEPROM for data storage, an ADXL326 $\pm 19g$ three-axis accelerometer, a multifunction ICSP/RS-232 interface, and an I/O port for analog data collection and digital control. The smallest flight computer packs these features onto a board that fits into a BT-5 sized rocket body tube (just over half an inch in diameter). The larger computers — sized to fit BT-20, BT-50, and BT-60 body tubes — add an I²C interface for external devices and a simple telemetry radio. Five of these computers fit onto one ExpressPCB MiniBoard (**Figures 1 and 2**). The BT-60 version uses all through-hole components, while the other versions use surface-mount parts.

The 14-pin PIC16F688 seems almost custom designed for an application like this. It's physically small (especially in its SOIC surface-mount package) and can draw less than 1 mA from a three volt supply, but still has 4K of program memory, several 10-bit analog-to-digital converters (ADC), an internal clock oscillator, and internal pull-up resistors on port A and MCLR which help minimize the parts count.

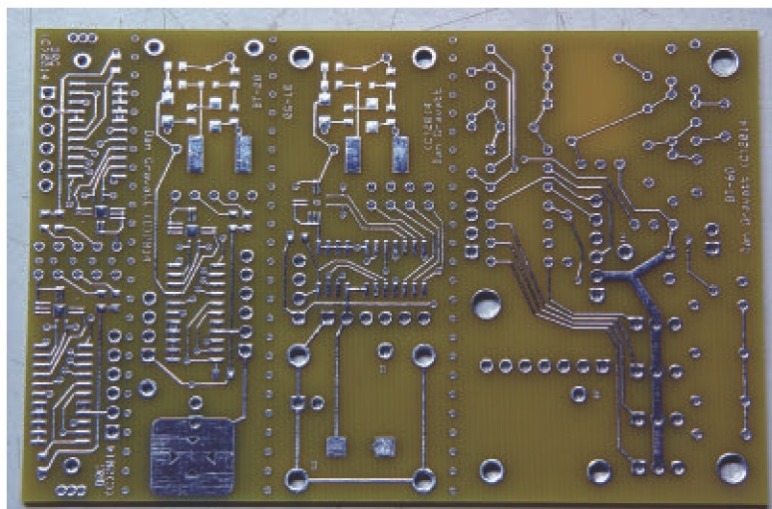
The schematic in **Figure 3** shows how the PIC interfaces with the rest of the components. Port A is used for ICSP, RS-232 data output, the I²C interface to the EEPROM and any external I²C devices, mode selection (flight mode or download mode), and switched power to the accelerometer and telemetry transmitter. Port C handles analog data collection from the accelerometer and external sensors, as well as digital I/O. **Figure 3** also includes a simple external interface circuit (J4, J5, U4, and associated components) and external five volt power source for use when downloading collected data to a PC via RS-232, if your PC's serial port (or USB to RS-232 converter) won't work reliably with the three volt signal output from the computer when using its onboard battery.

The telemetry transmitter is based on a design by Harry Lythall SM0VPO. Inductor L1 is 6.5 turns of bare solid 22 gauge wire wound around a 3/16" form with an overall length of about 1/2". You can stretch or compress the inductor a bit to tune the transmitter to the design frequency of about 144 MHz, so that crystal X1 can lock in that frequency in third overtone mode; 144 MHz is at the bottom end of the amateur radio two-meter band, so you can use a two-meter portable transceiver to listen to the telemetry. Be sure to follow Federal Communications Commission (FCC) rules for licensing and station identification.

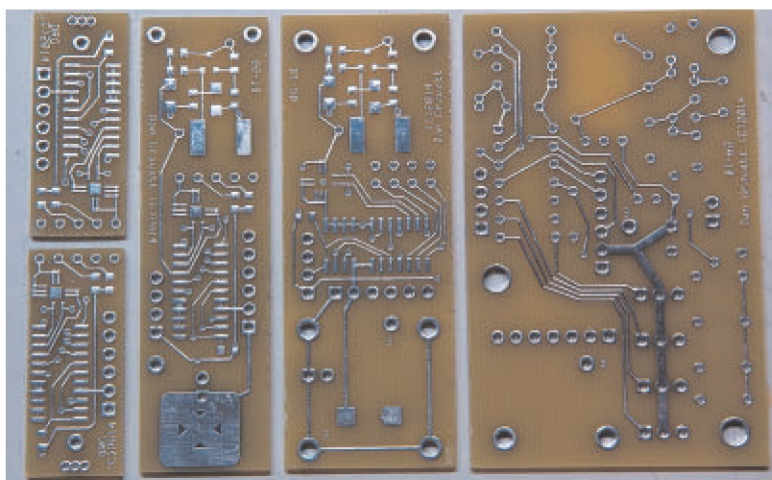
Alternatively, check out <http://transmitters.tripod.com/begin.htm> for component values that will yield a transmit frequency in the FM broadcast band and some possible variations on the design.

If desired, solder an antenna wire to the inductor one turn from the end connected to the positive supply voltage, being careful to route the antenna away from metal or other objects that may destabilize the transmitter. I've gotten a range of several hundred feet outdoors with a good quality receiver using no antenna wire at all.

The code uses pin RC4 to generate modulation signals for the telemetry radio, but as shown in **Figure 3** there is no direct modulation connection to the transmitter. Close proximity of the PIC I/O line to the transmitter provides a modest level of modulation; if desired, you can connect RC4 via a simple resistor-capacitor interface to the base of Q1 or to an external



■ **FIGURE 1.** ExpressPCB MiniBoard containing two BT-5s and one each of the BT-20, BT-50, and BT-60 flight computer boards.

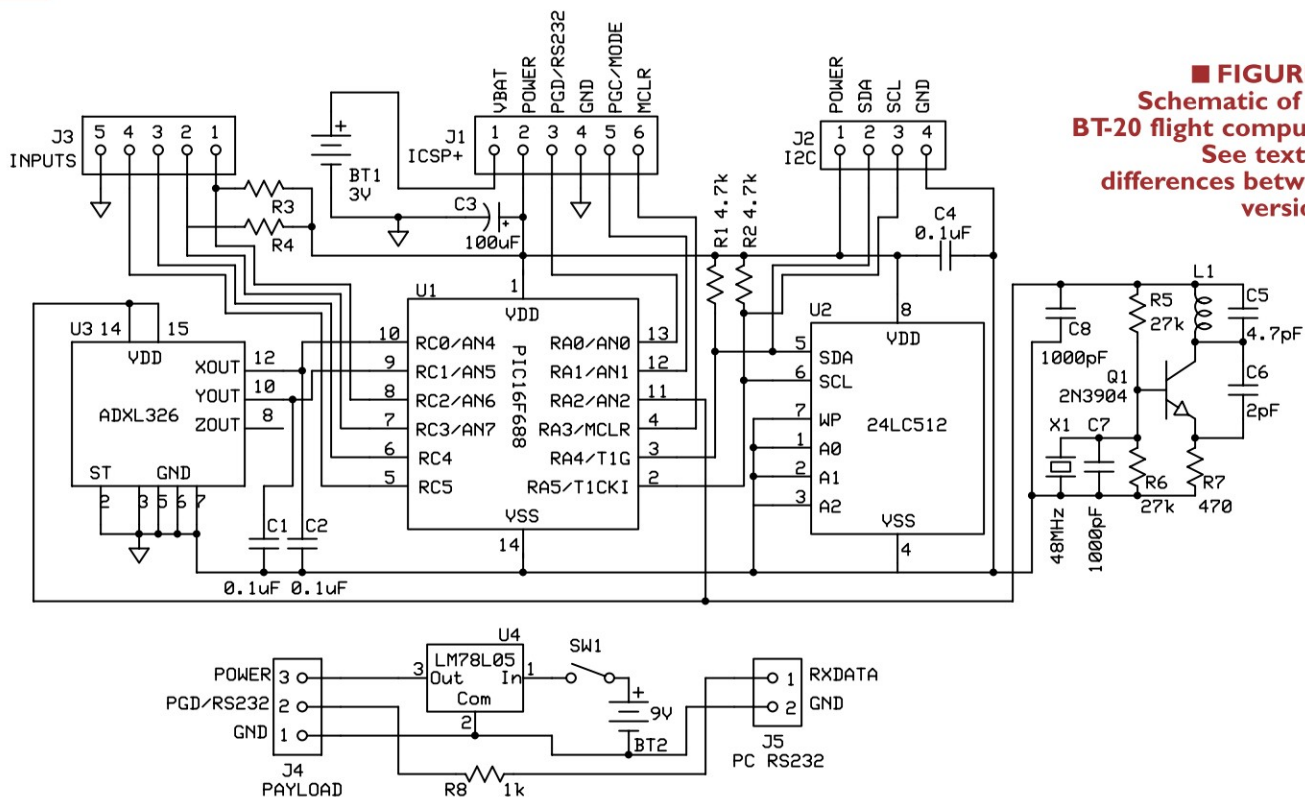


■ **FIGURE 2.** MiniBoard cut apart into individual flight computer boards.

transmitter for more precise modulation control.

Multifunction interface J1 serves several different purposes depending on whether you're programming the PIC, collecting data during flight, or downloading data to your PC. During flight, pins 1 and 2 are jumpered to provide power to the computer, and pins 4 and 5 are jumpered to set the computer to flight mode.

After a flight, the jumper is removed from pins 4 and 5 to set the computer to download mode, and pins 3 and 4 provide the RS-232 data (9600 baud, 8N1) to your PC via the external power/interface circuit described earlier. Pins 2 through 6 of J1 provide ICSP control of the PIC, but note that these pins are not in the standard ICSP order (a compromise to accommodate the other functions J1 supports), so you will have to make a simple adapter cable to your favorite ICSP programmer. The physical arrangement of the pin functions in J1 ensures that the computer can only be configured for one mode at a time.



■ **FIGURE 3.**
Schematic of the
BT-20 flight computer.
See text for
differences between
versions.



SF02 Laser Rangefinder

Fast and accurate distance measurements from 0-40 m (0 to 130 ft). Not affected by wind, noise, ambient light, terrain, air temperature, or changes in barometric pressure. **No calibration required** - ready to use right out of the box. #28043; \$349

Order www.parallax.com
or call 888-512-1024
(M-F, 8am-5pm, PDT)



The Convenient All-in-One Solution for Custom-Designed Front Panels & Enclosures



**FREE
Software**



ONLY \$90.24
with custom
logo engraving

You design it
to your specifications using
our FREE CAD software,
Front Panel Designer

We machine it
and ship to you a
professionally finished product,
no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

**FRONT PANEL
EXPRESS**

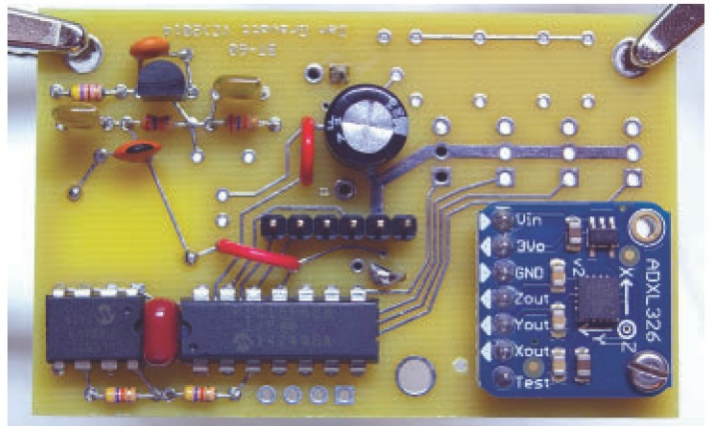
FrontPanelExpress.com
1(800)FPE-9060

Figure 3 shows the schematic specific to the BT-20 flight computer. The BT-5 flight computer differs from **Figure 3** in that it does not include I²C interface J2, filter capacitor C3, or the telemetry transmitter (transistor Q1 and associated components). The BT-50 flight computer differs from **Figure 3** only in that the accelerometer's X and Y connections to the PIC are swapped (see later discussion for why). The BT-60 flight computer has several differences from **Figure 3**:

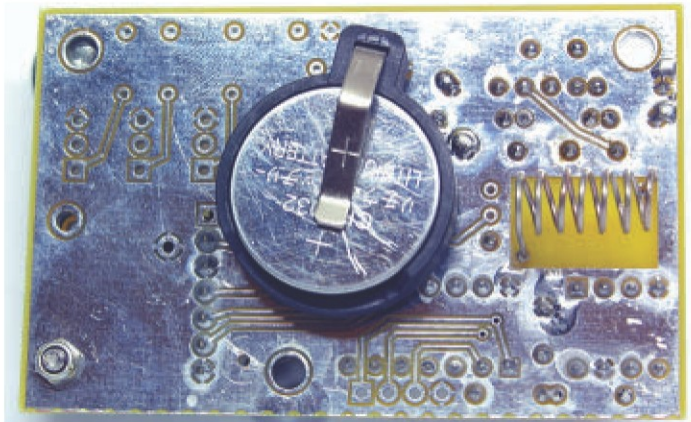
1. An AdaFruit ADXL326 breakout board (<http://learn.adafruit.com/adafruit-analog-accelerometer-breakouts>) holds the accelerometer and support components. Its onboard 3.3V regulator is unused, however.
2. The J3 analog and digital I/O port is in the form of four three-pin headers with power, ground, and signal to support servomotors. Each signal line provides for a pull-up or pull-down resistor for sensors if you want to use some or all of them for that purpose instead.
3. The Z axis accelerometer output is connected to the PIC, but this pin is shared with one of the three-pin I/O headers, so you can't use both functions at the same time.
4. The printed circuit board (PCB) includes a place for a three-terminal 3V regulator, so you can use a 6V main power source to power the servos and/or the telemetry transmitter. If you want to run the whole computer at 3V, just jumper the input and output pads for the regulator.
5. The PCB includes places for a 2N3904 transistor and base resistor to power the telemetry transmitter from 6V under control of pin RA2. If you want to run the whole computer at 3V, just jumper from the resistor pad on RA2 to the emitter pad of the transistor.

The code for the PIC (available at the article link) performs four main functions: configuring the PIC's internal peripherals and I/O pins for our purposes; collecting four channels of analog data (two from the accelerometer and two from other sensors) and saving them to the EEPROM; evaluating the accelerometer data as it is collected to detect motor burnout, parachute ejection and landing and taking actions based on these events; and downloading the data via RS-232 after the flight. The PICBASIC PRO code listing includes lots of comments and notes explaining these functions as well as other details not discussed here, so be sure to take a look.

In flight mode, the computer will collect four channels of analog data every 0.1 seconds and save them during the rocket's ascent. The rocket's X axis acceleration data is also loaded into an eight-word array variable FIFO style, and the code looks for patterns in the data that indicate



■ **FIGURE 4. Top of completed BT-60 flight computer showing AdaFruit accelerometer daughterboard.**



■ **FIGURE 5. Bottom of completed BT-60 flight computer showing battery holder and inductor L1.**

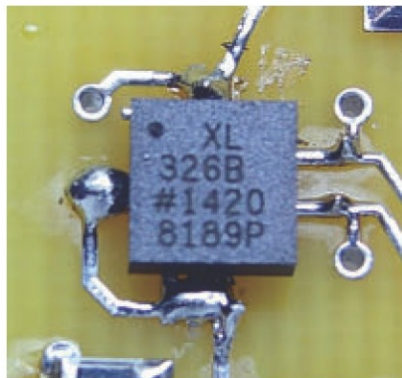
motor burnout (switch from upward acceleration to deceleration), parachute ejection (a period of deceleration followed by a sudden shock), and landing (a series of identical data points, assuming your rocket isn't swinging from a tree branch).

The threshold values in the code used to identify these patterns are somewhat generic and may need to be modified depending on the specific acceleration profile of your rocket. The code writes "markers" into the saved data when these events are detected to make it easier to identify them when processing the data. When parachute ejection occurs, the data collection rate changes to once every second, and a simple beep is sent by the telemetry radio. After landing, data collection stops, the PIC's ADC is turned off to save power, and the telemetry radio begins sending a series of beeps for radio location of your rocket.

Tight size constraints on the BT-5/BT-20 and BT-50 boards and the layout of the AdaFruit daughterboard on the BT-60 board prevented me from placing the accelerometer in the same orientation on each version of the computer. For all versions, I've defined the rocket's X axis to be vertical — whether or not this corresponds to the X axis of the accelerometer — and assumed that the



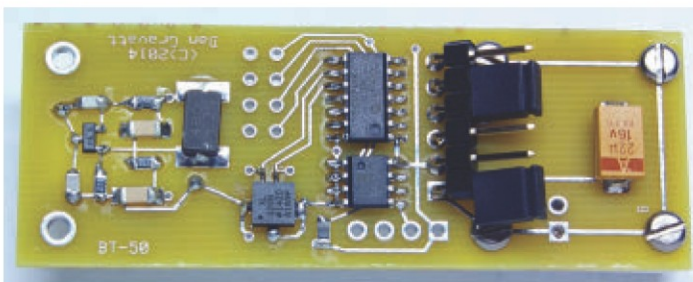
■ **FIGURE 6.** Close-up of pad array for accelerometer LFCSP, showing inner pad ends trimmed away.



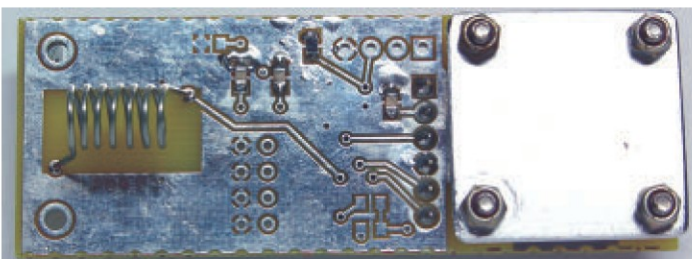
■ **FIGURE 7.** Close-up of accelerometer LFCSP, hand-soldered to the board.

computer will be installed in the rocket with its battery end down (or its accelerometer daughterboard end up for the BT-60 version).

The accelerometer chip is turned 90 degrees on the BT-50 version relative to the other versions, so I swapped the X and Y inputs to the PIC so the rocket's X axis data is routed to the same pin (RC1) in each version. For all versions, upward acceleration under thrust will result in vertical acceleration data values less than 512 (the accelerometer outputs an analog signal of half its supply voltage when under no acceleration, which the PIC's ADC converts to a number in the middle of its 10-bit range), which will then switch to data values greater than 512 after motor burnout as the rocket begins to decelerate. The data-parsing subroutines in the code assume these orientation and acceleration conventions in order to



■ **FIGURE 8.** Top of completed BT-50 flight computer.



■ **FIGURE 9.** Bottom of completed BT-50 flight computer, showing metal plate and screws securing the coin cell battery.

properly detect motor burnout and parachute ejection, though you can omit or modify them if you need to orient your computer differently in your rocket.

Construction

Construction of the BT-60 version of the computer with through-hole parts is very straightforward; the only challenge will be choosing from the various options for the power and I/O configuration described above. **Figures 4 and 5** show the top and bottom of the completed BT-60 computer. Construction of the three surface-mount versions is more challenging, though I deliberately picked relatively “large” versions of the parts to ease

the process as much as possible. There are plenty of resources on the Internet to teach you how to hand-solder these parts including the accelerometer chip, which is only available in a tiny LFCSP package. (Based on my experience, LFCSP stands for “a Little Frustrating, but you Can hand-Solder this Package.”)

However, I deliberately omitted a pad for the Z axis output of the accelerometer as I was not confident of my ability to solder it without bridging to the adjacent ground traces. Since the Y and Z axes are pretty much interchangeable in a rocket that may be spinning a bit during its ascent, this is not a problem.

Before you solder the LFCSP to its place on the board, you will need to remove a tiny bit of the inner parts of the traces so they don't short out through the central metal pad on the bottom of the LFCSP. I had to build the pad array for this chip from scratch as my version of ExpressPCB didn't have a pattern for it, and I wasn't able to position these pads any more precisely without compromising their alignment with the pads on the LFCSP. **Figure 6** shows the pads trimmed prior to soldering. I had good luck using a sharp hobby knife. **Figure 7** is a close-up of the LFCSP hand-soldered in place.

The battery installation methods for the three surface-mount computers are different. The BT-50 version uses a CR-2016, -2025, or -2032 lithium coin cell held in place by a small metal plate and four screws (**Figures 8 and 9**) since the coin cell holder used on the BT-60 simply won't fit inside a BT-50 tube when mounted to the PCB.

The BT-20 version is designed to use two LR-44 button batteries — one above and one below the PCB — connected and held in place via a stiff spring metal clip that wraps around the end of the PCB. Insulate the center part of the clip so it doesn't short out the batteries, and then use a large diameter piece of heat shrink to hold the batteries and clip securely in place.

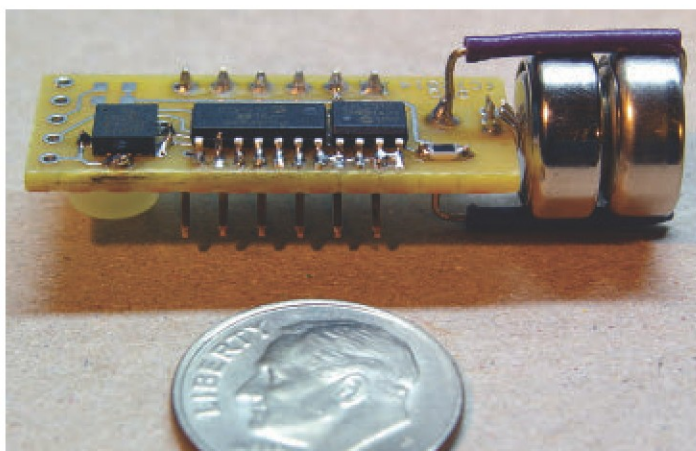
The BT-5 version is designed to use either two LR-44

batteries or one CR-1/3N lithium cell on the end of the board — connected and held in place by a contact soldered to the end of the board, and a spring wire loop from the board around the length of the battery or batteries (Figure 10). Insulate the sides of the spring wire loop if using two LR-44s so it doesn't short out the batteries, and then use a wrap of tape to hold everything securely in place.

Note that these unorthodox battery installations are only necessary if you intend to use any of the computers in the body tube diameter they are designed for. If you want to use, for example, the BT-5 version in a BT-50 diameter rocket, you can connect external power to the computer in any way that's convenient. The BT-20, -50 and -60 computer boards include mounting holes so they can be secured to a larger payload assembly in any diameter rocket. If you do this, you must ensure that the computer is oriented properly in the rocket so the accelerometer axes are correctly aligned, or adjust the code so that the computer knows which data stream corresponds to the rocket's X axis.

Customize It

I designed in a lot of flexibility and modularity so anyone could create a version that meets their specific



■ FIGURE 10. Completed BT-5 flight computer, showing details of battery holder construction.

PART #	ITEM
Surface-mount computers parts list (Digi-Key part numbers)	
PIC16F688-I/SL-ND	U1, Microcontroller, 14-SOIC
PIC16F1825-I/SL-ND	U1, Microcontroller, 14-SOIC (optional upgrade, see text)
24LC512-I/SN-ND	U2, EEPROM, 8-SOIC
ADXL326BCPZ-ND	U3, Accelerometer, 16-LFCSP
399-1170-1-ND	C1, C2, C4, 0.1 μ F 50V, 0805
478-6216-1-ND	C6, 2 pF 100V, 0805
712-1371-1-ND	C5, 4.7 pF 250V, 0805
478-1530-1-ND	C7, C8, 1,000 pF 50V, 1206
399-3769-1-ND	C3, 100 μ F 10V tantalum, size D
311-4.70KCRCT-ND	R1, R2, 4.7K, 0805
311-470ARCT-ND	R7, 470 ohm, 0805
311-27KARCT-ND	R5, R6, 27K, 0805
MMBT3904FSCT-ND	Q1, 2N3904, SOT-23
535-9853-1-ND	X1, 48 MHz crystal, 6 mm SMD
S1111EC-40-ND	J1, 40-position 0.1" 90 degree header
BT-60 computer parts list (Digi-Key part numbers except as noted)	
PIC16F688-I/P-ND	U1, Microcontroller, 14-DIP
PIC16F1825-I/P-ND	U1, Microcontroller, 14-DIP (optional upgrade, see text)
24LC512-I/P-ND	U2, EEPROM, 8-DIP
ADXL326	U3, AdaFruit accelerometer breakout board
296-29830-ND	Three volt LDO voltage regulator, TO-92 (see text)
XC943-ND	X1, 48 MHz crystal, HC-49
4.7KEBK-ND	R1, R2, 4.7K, 1/6 watt
27KEBK-ND	R5, R6, 27K, 1/6 watt
470KEBK-ND	R7, 470 ohm, 1/6 watt
490-7366-1-ND	C6, 2 pF 50V, ceramic
445-8586-ND	C5, 4.7 pF 50V, ceramic
445-5246-ND	C7, C8, 1,000 pF 50V, ceramic
493-1107-ND	C3, 100 μ F 50V, electrolytic
445-5258-ND	C1, C2, C4, 0.1 μ F 50V, ceramic
2N3904FS-ND	Q1, 2N3904, TO-92
S1011EC-40-ND	J1, 40-position 0.1" straight header
BC2032-E2-ND	BT-1, CR-2032 battery holder, PC mount

PARTS LIST

needs. Both the accelerometer and the telemetry radio are optional and can be omitted. Pretty much any type of I²C-based device can be added to the I²C port, and pretty much any type of resistive or voltage-generating sensor (temperature, pressure, humidity, light, sound, etc.) can be added to the analog inputs. Pads for voltage-divider resistors (R3 and R4) for these analog sensors are available in each version of the computer. Unnecessary analog inputs can be converted to digital I/O and vice versa.

There is plenty of code space left in the PIC for more sophisticated I²C communication, data analysis, telemetry data transmission, servo control, and digital I/O control subroutines. You can also replace the 16F688 with a 16F1825, which is pin-compatible with the '688 and adds peripherals including hardware PWM modules and a sophisticated data signal modulator, but you'll have to revise the code to use that chip.

However you decide to build it, this computer will satisfy your inner scientist at the same time you're putting on a good fire and smoke show for your spectators. **NV**

Soldering Reflow Controller

One of the more time-consuming tasks in my shop is building up surface-mount printed circuit boards (PCBs). Placing the solder paste on the board is somewhat tedious, and then standing over a reflow (or toaster) oven while carefully monitoring time and temperature adds to the task. Finally, a visual inspection of the finished assembly generally finds issues that need rework such as solder bridges, missed solder connections, and misaligned parts. The product being reviewed here helps to alleviate most of these problems.

I looked into using a homebrew controller for speeding up the cycle time for small surface-mount device (SMD) boards. I thought homebrew stuff could make the job of soldering SMD boards easier, but the problem I ran into is that many of them are supplied as an incomplete kit with some critical part (such as the power relay) missing. I imagine this releases the supplier from some liability since you cannot possibly use the kit without some modification. The other issue is that I had a batch of SMD boards that needed to be processed pretty quickly, so I ran out of time for another do-it-yourself project.

A soldering reflow oven and controller kit was generously provided to me by Beta Layout (www.reflow-kit.com) so I could try it out (**Figure 1**). Beta Layout also provides fabrication of blank PCBs, solder stencils, 3D board mockups, and custom front panels as part of their product line. The laser cut stainless steel SMD solder stencil is included free with a PCB order.

The Reflow Kit provided all the materials needed to automate my reflow soldering process. The toaster oven provides the heat source and safe environment for it. The controller (**Figure 2**) provides the timer function and cycle status to keep the user informed on the reflow process.



Figure 1. Reflow oven and controller.

The thermocouple provides the required temperature feedback to the controller. The controller is pre-programmed to run a specific temperature versus time profile that is well matched to the solder paste and SMD parts used. Over time, the solder manufacturers and parts vendors have determined the appropriate temperatures and dwell times needed to pre-heat the PCB and parts, activate the solder flux, reflow the solder, and cool down the PCB to minimize thermal stress on the board assembly.

There are different time and temperature profiles for different solder formulations such as lead and lead-free solders. The provided solder paste (**Figure 3**) is RoHS certified and is safer to use than leaded solder. There are cases where one is preferred over the other, but that is “solder fodder” for another time.

The solder stencil makes it a lot easier to apply the solder paste — I previously did this with a toothpick or medical syringe. It is messy and problematic to get the right amount of solder on a pad. It is really hard to control the 3D (or vertical) accumulation of solder on a pad manually. Think of it like trying to put a controlled quantity of sticky cookie dough on a spot; no two spots are ever the same size. Uneven solder application results in the



Figure 2. Beta Layout's reflow controller — US version.

tomb-stoning of two-legged parts (they stand vertically due to uneven surface tension of molten solder on the legs). Too much solder results in solder bridges between leads on integrated circuits of various packages. Too little solder results in pins or pads that are not connected between the traces on the PCB and the pads on the device.

These are especially hard to rework — if not impossible — on QFN and BGA parts. So, the stencil makes it much easier to apply the desired quantity of solder correctly in the first place, and will result in less rework after reflow soldering. Considering that the stencil is free with a PCB order, why not use it to your advantage. It is much more satisfying to do a quick visual inspection on a properly soldered PCB and then start testing assembled boards, than to have to fix tombstones and bridges. (Hey, that sounds like a new version of the snakes and ladders game!)

Opening Up the Box

A well packaged toaster oven with the reflow kit arrived in a single large box. Packed inside was a complete kit of everything I needed to get started (**Figure 4**). Contents included lead-free solder paste (which should be refrigerated to extend its shelf life), masking tape for stencil alignment, a solder squeegee/applicator, tweezers for component placement, a digital multimeter (with 9V battery, leads, and temperature probe), a PCB alignment kit, sample PCBs with an SMD parts kit, and solder paste stencil.

These items are part of the training opportunity which allows you to build a non-working PCB to test your ability to apply the solder paste, to identify and place the parts on the PCB, and finally to run the PCB through the reflow



Figure 3. Solder paste — this one contains no lead.



Figure 4. Reflow kit including tape, squeegee, digital multimeter with thermal probe, PCB alignment kit, tweezers, solder mask, practice PCB, and SMD parts.

The solder paste needs to be kept cold to prolong its shelf life. Putting it in the household fridge seems easy enough, but it is not a good idea. A small dormitory type fridge would be better, and make sure to label it accordingly.

My former workplace had a small fridge that was used to store solder paste and isopropyl alcohol. It was marked "solder and alcohol only." (Of course, it only took a couple days for the sign to be modified to read "lead and vodka only" ...)

The reflow oven should *never* be used for food once it has been used for soldering. It is common sense that we don't want chemicals, metals (heavy or otherwise), or surface-mount parts in our food. So, be safe and do not be tempted to use the reflow oven for food.

Others may see this nice looking toaster/counter-top oven and think it is just the thing for heating up their lunch. Without knowing its history, they will be taking a risk. I would suggest permanently marking the oven in some manner, stating that it is not safe for food.

Safety

25 YEARS **Beta**
LAYOUT
create:electronics

Create your own Solder Reflow station

NEW!

Reflow Controller:	\$ 315. ⁰⁰
Large Beta Reflow Kit:	\$ 178. ⁰⁰
Total Cost	\$ 493.⁰⁰

Laser Stencil for assembly **FREE**
with every PCB-POOL® order: \$ 0.⁰⁰
www.pcb-pool.com

For information, please call the
USA office toll free at 888-977-7443
Order online at
www.beta-eSTORE.com

eSTORE
Beta LAYOUT



Figure 5. The Create: Electronics dice roll kit.



Figure 6. Assembled dice roll kit with custom made front panel.

process. It is comforting to have the ability to practice some of these steps before you get down to work on your “live” circuit. Hopefully, you can work a few kinks out of the process before you get to those special parts you only have a few of.

The package also included a sample circuit called a “Beta Layout – create electronics kit” (**Figure 5**). This circuit includes the following: PCB, battery lead, SMD parts, 3D printed model of assembled PCB, laser cut SMD solder stencil, battery holder (also 3D printed), aluminum front panel (with printed graphics, machined holes, and mounting posts), and mounting hardware.

The “create electronics kit” really shows off the capabilities of Beta Layout, allowing one to build a



Figure 7. Thermal sensor attached to PCB in the oven.

professional kit with the included parts and front panel (**Figure 6**). This particular kit is for a digital dice roller circuit, but I could easily imagine it being a circuit/module of my own design! How would you like to have a professionally produced front panel for your latest project?

Initial Setup

Once everything in the package was unpacked, I started to wonder “can it be this easy?” It looked like it would be a simple “plug and go” operation, but I decided I better scan the instructions just in case. Hmm, what is this learn cycle?!

The learn cycle ends up being a fairly simple process. It turns out that the time required to bring the oven up to temperature varies across many different ovens. In order to adapt the controller to the oven’s heat-up time, you simply plug the provided thermal sensor into the controller, place the thermal sensor (attached to a PCB) into the oven (**Figure 7**), and start the controller with the “start” button. The rest is automatic. Check out **Figure 8** for these “learn” steps. The controller will turn the oven on, causing it to heat up. The controller then turns the oven off and measures the thermal over-shoot (how fast and how far the temperature climbs) so that it can more accurately control the desired temperature. With the learn cycle complete, it was time to solder.

User Experience

The controller does a great job of temperature and time control, and is much more accurate than my previous method of using a stop-watch and thermometer. This is primarily because it takes the human out of the loop for time and temperature regulation (**Figure 9**). The controller

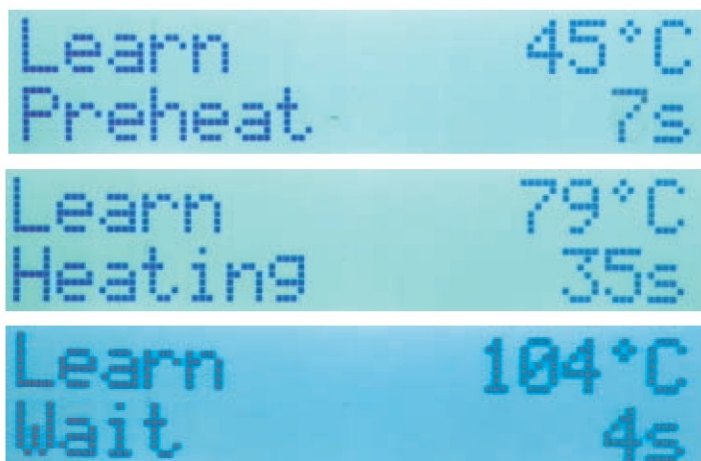


Figure 8. Status shown during the learn cycle.

also frees the operator to do other things while the cycle runs, such as prepare the next batch of parts or start checking the previous batch.

One of the neat things about the Beta Layout kit I used is the included stainless steel solder stencil (Figure 10). A solder stencil is usually a sheet of thin stainless steel or mylar which has cutouts for all of the solder pads on the PCB. By controlling the thickness of the solder stencil, you can place a known quantity of solder paste on each pad. This minimizes the chance of solder bridging or tombstoning parts since uniform solder paste distribution is almost guaranteed. The PCB alignment fixture consists of several L-shaped pieces of circuit board in two standard thicknesses. I chose the pieces that most closely matched the thickness of the PCB in the kit I was working with. Refer to Figure 11.

The larger L-shaped PCB is attached to a piece of hardboard or Masonite type material with the provided masking tape. The kit PCB is then placed in the inside corner of the L. A smaller L-shaped PCB is used to hold the opposite corner of the kit PCB in place and is affixed with tape. The stencil is then manually aligned to match the solder pads on the kit PCB. Once the stencil is aligned, it is taped to the fixture along one edge with the provided masking tape. Taping along only one edge allows the solder stencil to be lifted safely away from the PCB once the solder paste is applied.

Let's look at how we get solder paste onto the PCB. The solder paste comes in a 100 gram container that should be kept refrigerated to prolong its shelf life as mentioned previously. This particular paste is lead free. The solder paste should be taken out of cold storage and brought to room temperature before use. The paste should also be stirred for 15 seconds or so before being applied. I used a disposable plastic knife for stirring and applying the paste to the stencil.

The solder paste is applied by putting a generous bead along the edge of the stencil near the masking tape



Figure 9. Status display of a solder cycle in process.

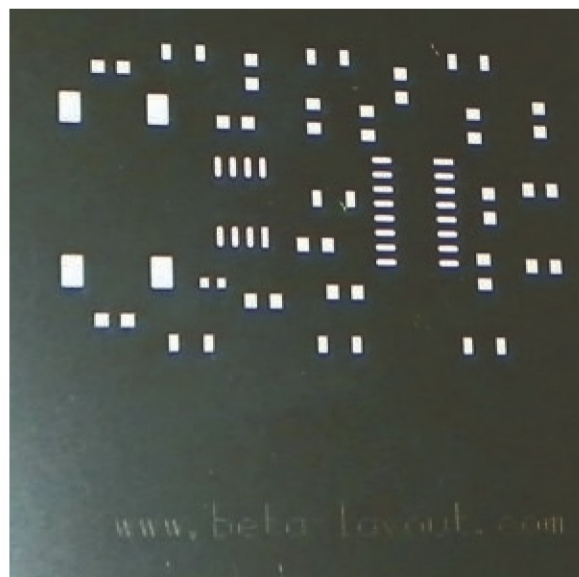


Figure 10. Stainless steel solder mask.

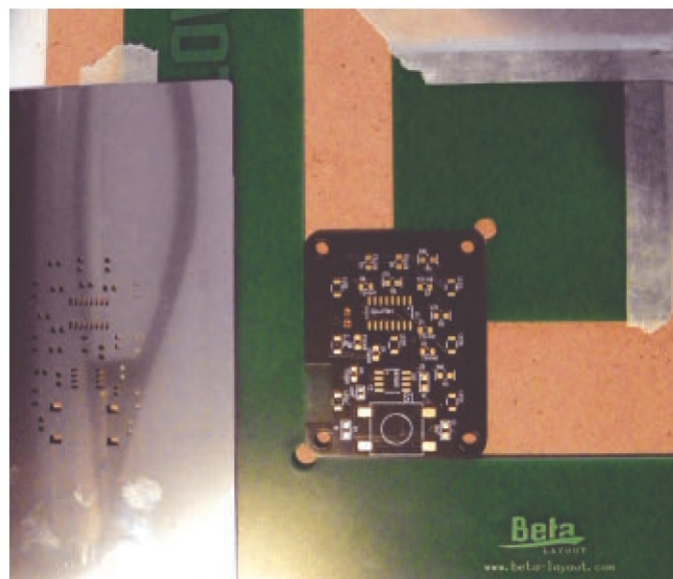


Figure 11. Board held in alignment fixture, between two L-shaped green PCBs.

(Figure 12). Since the solder paste is thick and sticky, the “bead” of paste looks more like a small mountain range. The included stainless steel squeegee/spatula is then used to scrape or spread the solder paste across the stencil, thus depositing a layer of paste into each opening of the stencil (Figure 13). Once the solder paste is spread over each stencil opening, the stencil can be flipped up to



Figure 12. Solder paste applied to mask, but not yet spread out.

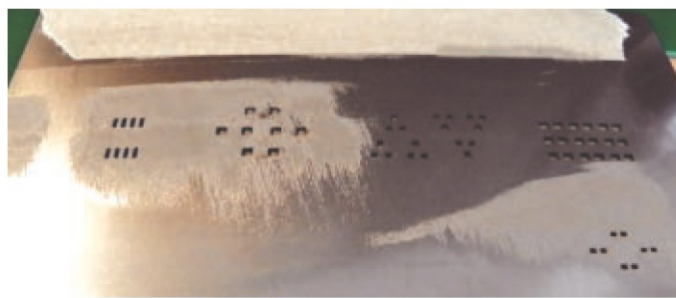


Figure 13. Solder paste spread across the stencil.



Figure 14. Solder stencil flipped up after paste application.



Figure 15. Solder paste applied to the pads on the board.



Figure 16. SMD components placed on PCB pads with solder paste.

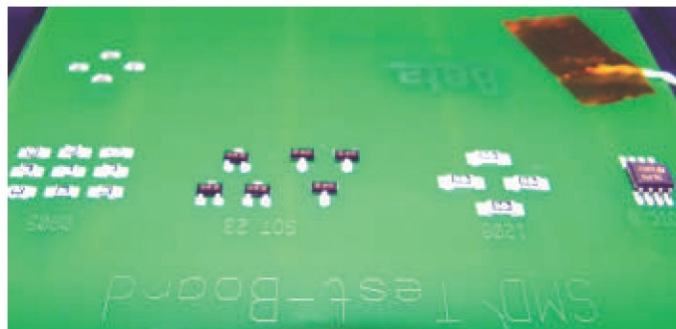


Figure 17. Board placed in oven with thermal sensor attached.

reveal the PCB with solder paste accurately placed at a uniform thickness (**Figure 14**). **Figure 15** shows this even distribution of solder paste.

The next step is to put any excess solder paste back into the original container and to clean up any residual paste. Be careful not to put any contaminating material into the solder paste as it could affect the spreadability in subsequent applications. I used a paper towel and some isopropyl alcohol to clean the stencil and the spatula. I noticed that Beta now sells a wet wipe product for this purpose.

Now for the fun part: placing the surface-mount components on the board. I used the included tweezers and a finer tipped pair of tweezers I already had to remove the SMD parts from their paper tape and plastic carriers, and place them carefully on the PCB. Orientation

of polarized parts such as LEDs, diodes, and capacitors is important. The solder paste is sticky enough to hold the components in place. You just have to be careful not to bump any components while reaching towards the center of the board to place some other part. **Figure 16** shows the components placed on the solder paste.

Once all the parts are placed, it's time to put the board in the oven and power up the controller. I attached the thermal sensor to the test PCB for the first run; thereafter, I attached it to a similar sized PCB (also placed in the oven) so that it senses the same temperature as the kit PCB. I chose to use high temperature polyamide tape to affix the thermal sensor to the board (see **Figure 17**). I did find that the tape was less sticky after a few cycles, so I also used a piece of wire inserted through the PCB to hold it in place on later runs.

With the thermal sensor attached and the kit PCB placed in the oven, I closed the oven door, powered up the controller, and started the cycle (**Figure 18**). The first thing I noticed was a couple of yellow LEDs at the bottom of the controller. Yellow and red LEDs make me a bit nervous, but these were only showing the status of the USB connection to the controller (**Figure 19**). A USB connection is not needed to use the controller, but it is provided to allow firmware updates, etc.

Temperature calibration, temperature units (degrees Celsius or Fahrenheit), backlight brightness, and many



Figure 20. "To Hot" error message.

other parameters can be changed via the keypad or via the USB interface. You can also adjust the time, end temperature, and power rating of the oven for each phase of the reflow cycle, as well as emulate the controller keypad via this interface.

Pressing the start button gets the process under way. The controller provides status info on a two-line backlit LCD display giving the cycle name, time, and temperature so that you will know how things are progressing. For repeat cycles, the controller will not allow you to start another cycle until the thermal sensor cools to about 45°C (Figure 20). The warning says "To Hot" which does mean that the oven is too hot. This is reasonable since the PCB needs to be cool enough to handle so you can remove it from the oven before loading in a new board.

At the end of the cycle, the controller beeps to let you know it's complete. Figure 21 shows the status display at the end of the cycle — note the elevated temperature reading. At this point, you should open the door to allow the oven to cool. (I found myself wishing the door would open automatically which is an unreasonable request. After watching the oven like a hawk for the first few cycles, it worked so well and so predictably that I would be off doing something else when the end-of-cycle beep occurred.) This is a huge time saver over monitoring time and controlling temperature manually. Of course, it's always a bit exciting to watch and monitor every detail of



Figure 18. Controller ready to start the cycle.

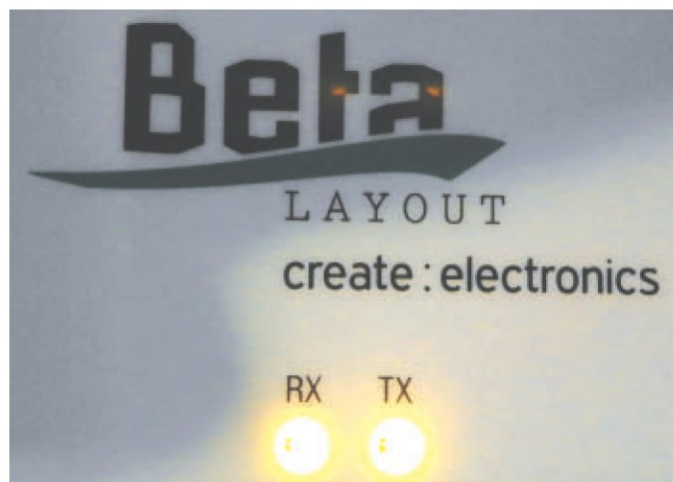


Figure 19. USB indicators. They illuminate when the controller is on.



Figure 21. End of cycle indication. High temperature is the clue that this is the end of the cycle.

a new process. I did carefully observe the first time I used the reflow controller, noticing the stages the controller went through, the time of each cycle, watching through the glass door of the oven as the solder started to melt, and opening the door at the end of the cycle.

NEW! from **velleman**



(Sold as kit)

3D PRINTER KIT

Model K8200

Advanced kit requires mechanical, electronics and computer skills

Build-it-yourself 3D printer. Prints objects of maximum size of 20 x 20 x 20 cm using either PLA or ABS filament (3mm.) Extremely fast, reliable and precise even when printing at high speeds. Model K8200 is compatible with RepRap software and firmware (Free download). Sturdy aluminum construction, with heated print bed. Nine different colors of PLA filament available at \$39.00. Complete with power supply and all necessary parts. Filament sold separately.

NOW ONLY \$795.00

Part No. 32VKK8200M





Electronix EXPRESS

TERMS: Min. \$20 + shipping. School Purchase Orders, VISA/ MC, Money Order, Prepaid. NO PERSONAL CHECKS, NO COD. NJ Residents: Add 7% Sales Tax.

In NJ: 732-381-8020
FAX: 732-381-1006

900 Hart Street • Rahway, NJ 07065
800-972-2225

<http://www.elexp.com>
email: electron@elexp.com

Other Services

The Create: Electronics kit included some examples of other services that Beta Layout provides. Along with the PCB and a complete kit of parts, there were some other useful items. One minor grumble — the included solder stencil had the website printed on the bottom side. This confused me for a minute because I expected the text to be visible from the side the solder would be applied to.

On the positive side, the stencil is free with the PCB and it is a very useful addition!

The kit also included a 3D printed version of the populated printed circuit board, a 3D printed battery holder, and a custom printed and machined front panel. Check out **Figures A, B, and C**.

The front panel had threaded standoffs installed, holes drilled, and custom graphics printed on the top. Each of these extra features will help to put the finishing touch on a professional looking project.

You may want to take a look at their website at www.beta-layout.com to see some examples of their capabilities. They will help make your next project stand out.

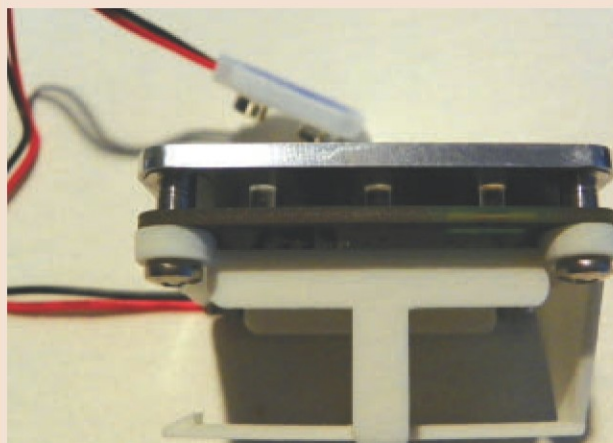


Figure B. End view showing the assembled dice roll kit.

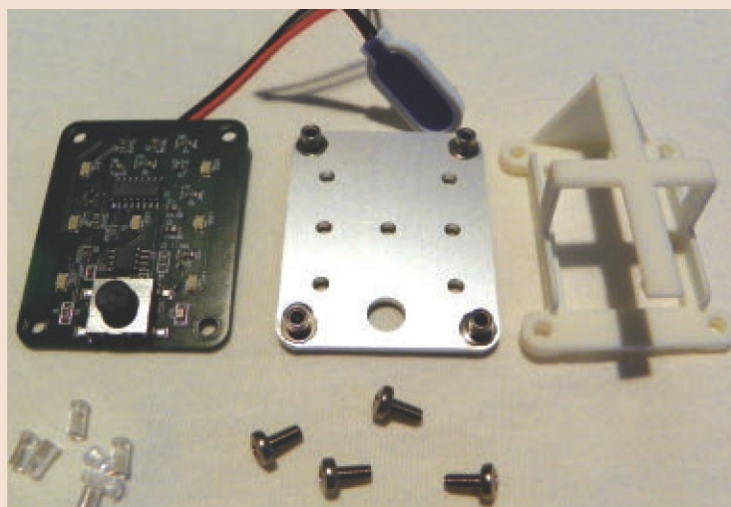
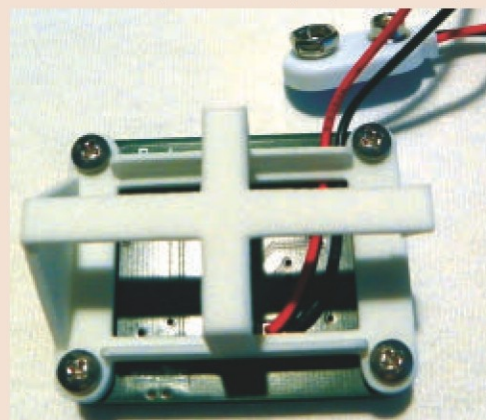


Figure A. Parts included with the dice roll kit. PCB (after fabrication) is on the left. Front panel with machined holes and threaded standoffs is in the center. 3D printed battery clip is on the right.

Figure C. Rear view showing the battery holder and nine volt battery clip.



I usually am looking to see if the solder melts completely and if the surface-mount parts move. The solder typically does not all melt at the same instant due to hot and cool spots in any oven. It will start melting in one area and then melt fairly quickly at spots across the entire board. Another thing I watch for is parts self-aligning to the pads of the PCB. The surface tension of the molten solder will usually pull surface-mount parts into better alignment with the pads. The last things I look for (and hope not to see) are problems such as parts that stand on end (tombstone), solder bridges (multiple pins connected due to too much solder), or parts that float on the solder and don't sit down evenly. (Floating is more likely to happen with QFN or BGA parts when the solder is not evenly applied to the pads.)

Conclusion

The controller and oven combination work very well

in automating the tedious tasks of timing and temperature control for solder reflow operation. I am sure the controller would work well with almost any toaster/counter-top oven that could provide the appropriate temperature, since the included thermal sensor and the learning cycle will “teach” the controller the appropriate heat-up characteristics for the oven.

Since the controller and thermal sensor provide a closed-loop control system, it truly does take the tedium out of the process and makes controlling the soldering temperature profile a one-touch affair. The solder stencil (if you happen to have one for your board) makes the solder application much easier and almost guarantees your success versus applying solder paste by hand and having to fine-tune solder paste volume and placement.

All in all, I see this as being a worthwhile addition to a hobbyist or small business workshop. It will definitely save time, and make that time on the workbench more productive. **NV**

The Arduino Classroom

Arduino 101 — Chapter 12: Data Logger/Part 1

We have come a long way with our Arduino 101 study. We've learned how to install the Arduino IDE (integrated development environment) software and use it with the Arduino UNO hardware. We've learned about electric circuits and how to use that knowledge to build devices on breadboards. We've learned about digital output and how to use it to control an LED. We've learned about digital input and how to use it to read the state of a pushbutton. We've learned some of the basic fundamentals of computer programming by writing programs that sense a pushbutton and turn an LED on or off. We've learned about analog output by generating variable voltages using PWM, and we used this to vary the brightness of an LED. We've learned about analog input by measuring voltages with an ADC (analog-to-digital converter). We used those measurements to learn about Ohm's Law, and we applied that to controlling the angular motion of a servomotor. We learned about displaying information using LEDs, including displaying characters with a seven-segment LED. We learned about keeping time with a computer, how to set our Arduino time accurately using Processing on a PC, and we displayed that time on a seven-segment LED. We learned how to sense light and temperature, and finally, we learned how to make sounds on a piezo speaker.

Yes, we have come a long way and learned a lot about the basics of computing and electronics. In this chapter, we are going to learn how to store data on our Arduino UNO so that we can design a battery-operated device that can run without being connected to a PC, and can collect environmental data that can later be uploaded to a PC for analysis. This will be our Arduino 101 capstone project: an Arduino data logger. It will help reinforce much of what we have learned so far, and let us demonstrate that knowledge. Plus, it will provide a solid basis for moving on to Arduino 102: A class that will teach the fundamentals of robotics.

In Memoriam: Jay Flanders

My best friend, Jay Flanders passed away while I was writing this chapter. Jay edited all the chapters before this one, and I'm sure that this chapter and all my work from this point forward will be less from the absence of his wise input. We will never again talk about nerdy stuff like the Arduino while having salads at McDonalds. He had a great laugh and for me, a light has gone out of the world. I dedicate the Arduino 101 series to him.

What is a Data Logger?

A data logger records data over time in a specific location. There are many examples of data loggers. They may be included in shipping containers for perishable goods that must be kept at a certain temperature. A time and temperature measuring data logger is started in the factory before the container ships, and then after the shipment is received the data logger can be queried to determine the temperature changes while in transit.

Another example is the Holter monitor: A device that can be carried by a medical patient and records heart rhythms over 24 hours. You may see data loggers used in remote unattended weather stations; these can include offshore buoys that are left to drift for months. There are numerous applications for data loggers.

What is EEPROM Memory?

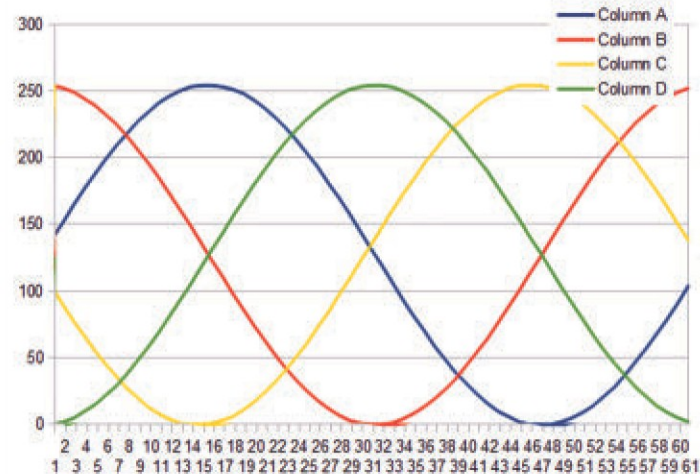
As you may have learned the hard way, when the Arduino loses power it also loses the values you've stored in variables. Back in Chapter 7, we used the PC to set

calibration variable values for the potentiometer angle. We recorded the ADC value when the pot was pointed at zero, and recorded it again when pointed to 180. Based on those readings, we would then map the ADC values to angles between 0 and 180 degrees, and then store these values in volatile memory ('evaporates' when power is removed). Each time we re-powered the Arduino, we had to calibrate it again.

We somewhat alleviated that problem in Chapter 9 when we learned to use the Arduino with a battery that continued to provide power to the volatile memory after the USB connection was removed, but batteries get unplugged or eventually run out of juice. So, how do we get the Arduino to remember things when the power goes off? We use the EEPROM (Electrically Erasable Programmable Read Only Memory) which is a form of non-volatile memory (doesn't 'evaporate' when the power is removed) that remembers the data until it is reprogrammed with new data. The Arduino UNO has 1,024 bytes of EEPROM. This memory is incredibly simple to use, plus we are provided with a library of functions that let us write or read bytes to addresses.

Keeping a Timestamp for Data Sensing

Well that's great, if you want to store eight-bit byte variables (numbers between 0 and 255). Unfortunately, there is a slight complication if you want to store other types of data that we've been using. The values from the ADC are 10-bit and may be from 0 to 1023. However, we store these values in 16-bit integers. As we saw in Chapter 9, Unix timestamps are 32-bit long integer values. We need an algorithm for converting these data types into bytes for storage and then retrieving those bytes and restoring the original data type. Since the ADC values are stored in 16-bit *ints*, we can convert them into two eight-bit bytes for storage. For the time value, we can convert and store 32-bit long integers as four eight-bit bytes.



■ FIGURE 1: Raw data shown in graph.

Conversions between these data types then become some relatively simple computing operations.

To keep things even simpler, we will use a special library (storeEEPROM) that automatically stores and retrieves 16-bit *ints* and 32-bit long *ints* in EEPROM. You can find this library in the a101_ch12_supplemental.zip file at the article link.

If you want to see how these algorithms work, you can look at the original source code for each function. The functions in that library are:

```
void storeIntEEPROM(int myAddress, int myInt)
int retrieveIntEEPROM (int myAddress)
void storeLongEEPROM (int myAddress,
    long int myInt)
long int retrieveLongEEPROM (int myAddress)
```

By now, you should be able to read each function description above and see what it does and how it is used. For instance, the first function stores a 16-bit *int*, *myInt* at the EEPROM location *myAddress*. The second function will retrieve a 16-bit *int* from the *myAddress* location. We will install and test this library in Lab 2.

Charting Arduino Data on a PC

Now that we know how to store raw data in an EEPROM, let's look at how we may show that data. We might write a program that reads four sensors once per second, and stores a data byte for each sensor in the EEPROM in blocks of four bytes each second. When we have finished reading that data after say, a minute, we may then send that data to a PC for interpretation. This sort of data is shown here with raw comma separated data:

```
0,152,252,89,2,165,249,77,5,177,244,66,10,188,239
,55,15,199,232,45,22,209,225,36,29,218,216,27,38,
227,206,20,48,234,196,13,58,241,185,8,69,246,173,
4,81,250,161,1,93,253,149,0,105,254,136,0,118,254
,123,1,131,253,111,3,143,251,98,6,156,248,86,11,1
68,243,74,17,180,237,63,24,191,230,52,32,202,222,
42,41,212,213,33,51,221,203,25,61,229,193,18,73,2
36,181,12,84,242,170,7,97,247,158,3,109,251,145,1
,122,253,132,0,134,254,120,0,147,254,107,1,160,25
3,94,4,172,250,82,8,183,246,71,13,195,241,59,19,2
05,235,49,26,215,228,39,34,224,220,31,44,231,210,
23,54,238,200,16,64,244,190,10,76,248,178,6,88,25
2,166,2,100,254,154,0,113,254,141,0,125,254,129,0
,138,252,116,2,151,249,103,5,163,245,91,9,175,240
,79,14,187,233,67,21,198,225,56,28,208,217,46,37,
217,207,37,47,226,197,28,57,233,186,21,68,240,175
,14,79,245,163,9,91,249,150,5,104,252,138,2,
```

When we list that data, we see that it doesn't make much sense to us. What is actually going on with these four sensors? We can put this data in a spreadsheet and have a chart drawn for the data showing the output of each of the four sensors as shown in **Figure 1**. Now, the data makes sense to us; we are seeing data that represents four sine waves.

We will learn how to do this in Lab 6 using the free Google Chart browser application.

Lab 1: Storing and Retrieving Byte Data in EEPROM

In this lab, we will learn to use the Arduino EEPROM functions. To use this library — which is included with the Arduino distribution — you must add `#include <EEPROM.h>` at the top of the source code file.

Parts required:

1 Arduino
1 USB cable
1 PC

Estimated time for this lab: 10 minutes

Check off when complete:

- ☐ We will only use the Arduino board for this lab.
- ☐ Load in the following program into the Arduino IDE. All program files are available at the article link.

```
//A101_ch12_EEPROM_library_test 11/13/14
#include <EEPROM.h>

void setup(){
  Serial.begin(57600);
  Serial.println("EEPROM library test rev 1.0");

  // store and retrieve two bytes
  Serial.println("Store 8-bit byte 123 at
address 0");
  EEPROM.write(0,123);

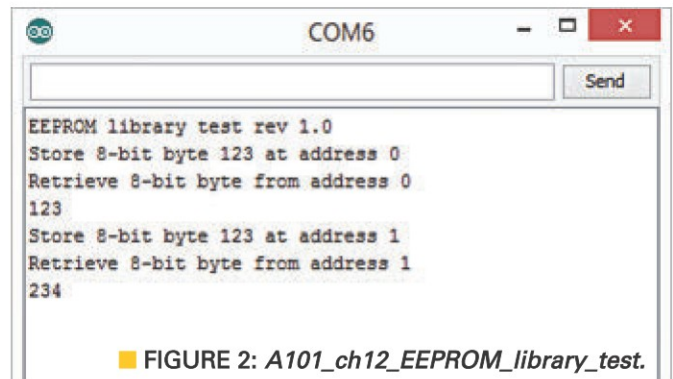
  Serial.println("Retrieve 8-bit byte from
address 0");
```

```
byte myByte = EEPROM.read(0);
Serial.println(myByte);
Serial.println("Store 8-bit byte 123 at
address 1");
EEPROM.write(1,234);

Serial.println("Retrieve 8-bit byte from
address 1");
myByte = EEPROM.read(1);
Serial.println(myByte);
}

void loop(){
  // do nothing
}
```

- ☐ Compile and run the program.
- ☐ Open the serial monitor and verify that you get results as shown in **Figure 2**.



■ FIGURE 2: A101_ch12_EEPROM_library_test.

Lab 2: Storing and Retrieving Non-byte Data in EEPROM

In Lab 1, we used the standard Arduino EEPROM library to store and retrieve eight-bit byte data. We will now use a custom library written for Arduino 101 that lets us store 16-bit *int* and 32-bit long *int* data types. This will allow us to store 10-bit ADC values as *int* data and to store Unix timestamps in long *int* data. In this lab, we will install and test the *storeEEPROM* library.

Parts required:

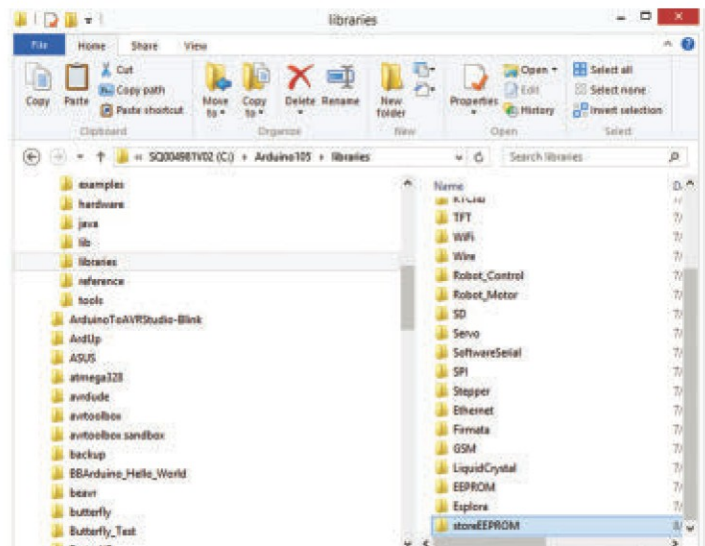
1 Arduino
1 USB cable
1 PC

Estimated time for this lab: 15 minutes

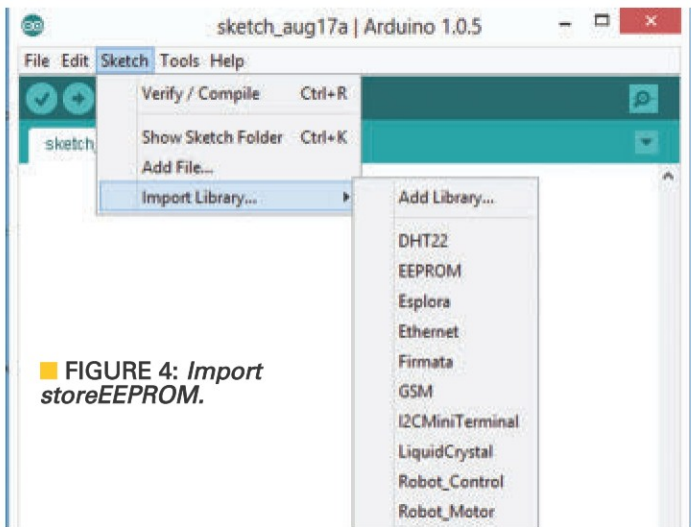
Check off when complete:

- ☐ We will only use the Arduino board for this lab, however, you should leave the most recent circuits on the Arduino proto shield since the program will not affect it and it will be used again in the next lab.
- ☐ Get the *storeEEPROM* file at the article link.
- ☐ Next, find your Arduino directory (usually in drive C; for instance, I have Arduino 1.0.5 located at C:/Arduino105).

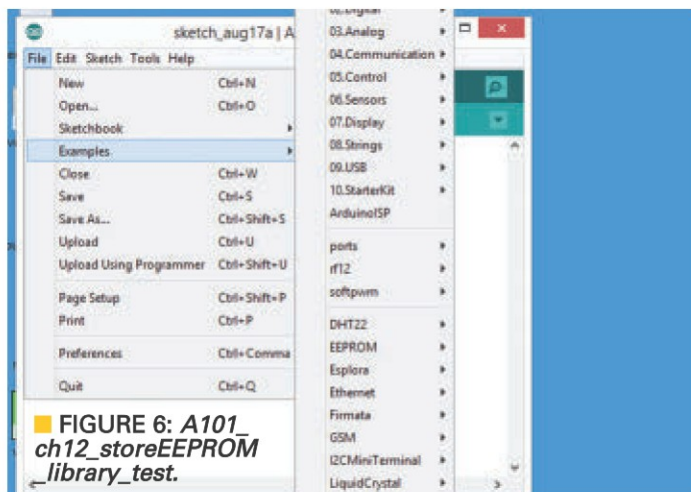
- ☐ Unzip the *Arduino_101_Ch12_January_2015_supplemental.zip* file and copy the *storeEEPROM* directory to the Arduino libraries directory. **Figure 3** shows my Windows Explorer with the file copied to the correct directory. It must be in the libraries directory for the Arduino IDE to find it.



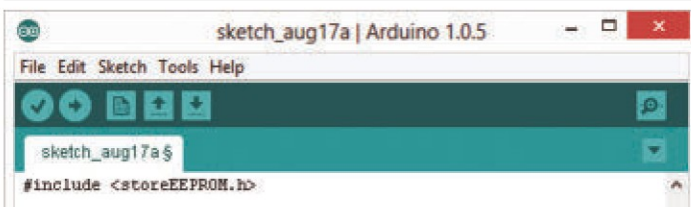
■ FIGURE 3: Put *storeEEPROM* in the libraries directory.



■ FIGURE 4: Import *storeEEPROM*.



■ FIGURE 6: A101_ch12_storeEEPROM_library_test.



■ FIGURE 5: Add *storeEEPROM* header.

- Next, verify that the Arduino IDE can find the *storeEEPROM* library. Open the Arduino IDE and then click on the Sketch/Import Library menu item. You should see the *storeEEPROM* library as shown in **Figure 4**. This verifies that you have successfully added the library to the Arduino directory.
- Click on *storeEEPROM*; it will add `#include <storeEEPROM.h>` to your source code as shown in **Figure 5**. You are now ready to write code that will use the Time library. Note that you aren't required to use this technique to add the header. You can manually type it in yourself, but importing it as discussed above proves that it is in the correct location for the Arduino IDE to find it.
- Open the File/Examples/*storeEEPROM*/Examples/*storeEEPROM_test* as shown in **Figure 6**. This file records an *int* and a *long* (16-bit and 32-bit data types) into the eight-bit EEPROM and then reads them back to verify that the library functions work as advertised.
- The source code from the example is as follows:

```
//A101_ch12_storeEEPROM_library_test 8/16/14
#include <EEPROM.h>
#include <storeEEPROM.h>

void setup(){
  Serial.begin(57600);
  Serial.println("storeEEPROM library test rev 1.0");

  // test int
  Serial.println("Store int 12345 at address 0");
  storeIntEEPROM(0,12345);

  Serial.println("Retrieve int from address 0");
  int myInt = retrieveIntEEPROM(0);
  Serial.println(myInt);

  // test long
  Serial.println("Store long 1234567890 at address 2");
  storeLongEEPROM(2,1234567890);

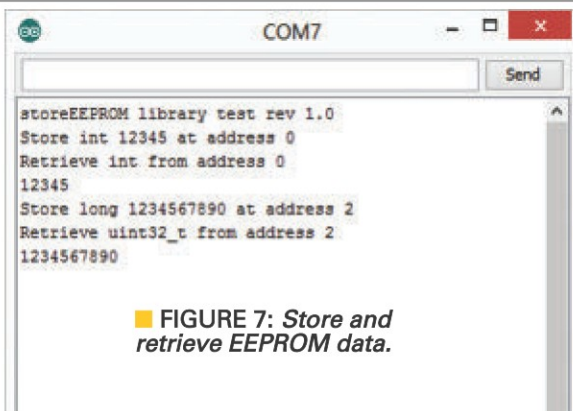
  Serial.println("Retrieve uint32_t from address 2");
  long myLongInt = retrieveLongEEPROM(2);
  Serial.println(myLongInt);
}

void loop(){
  // do nothing
}
```

- Compile and upload the code.
- Open the serial monitor and verify that you see the results shown in **Figure 7**.

Lab 3: Store Calibration Data

In Lab 3 of Chapter 7, we built a potentiometer dial that let us sense angles from zero to 180 that were set by rotating the potentiometer and reading the resulting voltage with the Arduino ADC. In that lab, we had to



■ FIGURE 7: Store and retrieve EEPROM data.

calibrate the dial each time we cycled the Arduino power because the calibration values were stored in SRAM (volatile memory) that forgets the data when the power is off. In this lab, we will learn how to calibrate the dial and store the calibration value in EEPROM.

Parts required:

- 1 Arduino
- 1 USB cable
- 1 Arduino proto shield and jumper wires
- 1 Potentiometer
- 1 100 Ω resistor
- 1 Pot dial angle image

Estimated time for this lab: 30 minutes

Check off when complete:

- ☐ This lab uses the potentiometer (pot) dial shown in Lab 3 of Chapter 7.
- ☐ Assemble the potentiometer dial and pointer and plug it into the Arduino proto shield breadboard as shown in **Figures 8 and 9**.
- ☐ Load the following program into the Arduino IDE:

```
// A101_ch12_EEPROM_dial_calibration Joe Pardue
// 11/12/14
// This program lets us store two int sized
// variables that can be used to calibrate the
// voltage from a potentiometer to coorespond
// to the angle shown on the dial

#include <EEPROM.h>
#include <storeEEPROM.h>

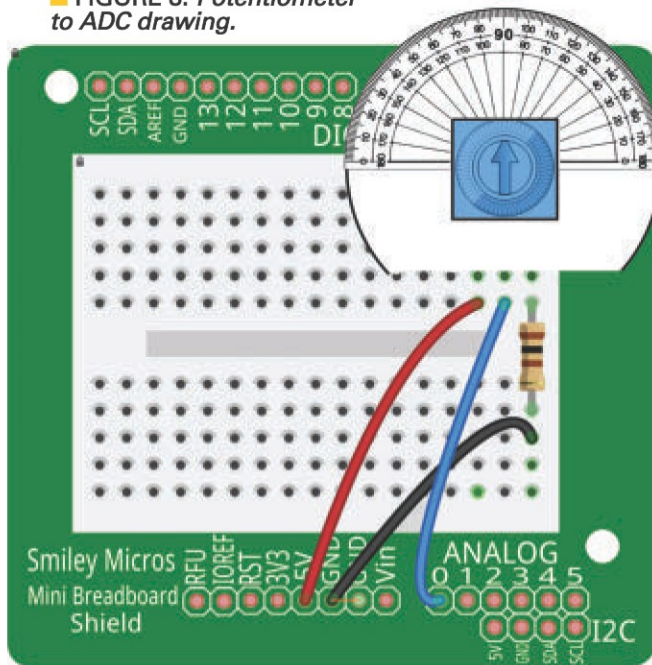
// EEPROM address to store the calibration
// values
// NOTE we are storing ints which take 2 bytes
// memory
#define ZERO_ADDRESS 0
#define ONEEIGHTY_ADDRESS 2

int sensorPin = A0; // analog input pin
int zero = 0; // calibration reading for 0
// degree
int oneeighty = 0; // calibration reading for
// 180 degree

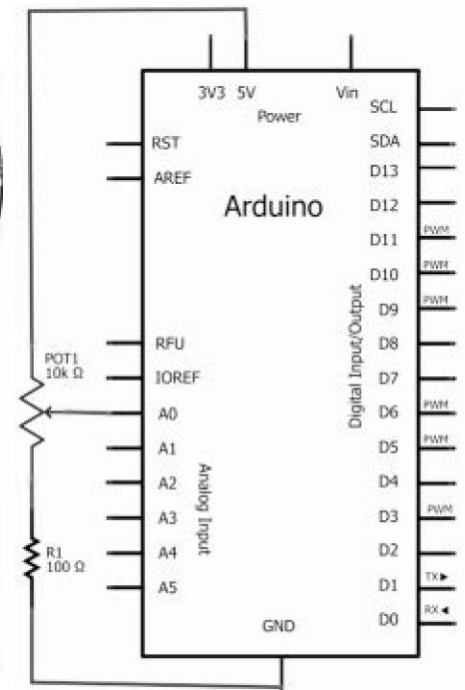
void setup() {
  Serial.begin(57600);

  Serial.println("A101_ch12_EEPROM_dial_calibration
1.0");
  Serial.println("Input 'z' to set zero");
```

■ **FIGURE 8: Potentiometer to ADC drawing.**



■ **FIGURE 9: Pot to ADC schematic.**



```
Serial.println("Input 'o' to set oneeighty");
Serial.println("Input 's' to show the
values");
Serial.println("Input 'r' to read the angle");
showCalibration();
}

void loop() {
  if(Serial.available())
  {
    char c = Serial.read();
    if(c == 'z') // get the zero degree
                  // calibration value
    {
      // read the value from the sensor:
      // analogRead returns an int so we divide
      // by 4
      // to get a byte sized result (induces
      // some error)
      zero = analogRead(sensorPin);

      // put zero in EEPROM
      storeIntEEPROM(ZERO_ADDRESS, zero);

      // Report the setting
      Serial.print("You set 0 degree to: ");
      Serial.println(zero);
    }
    if(c == 'o') // get the 180 degree
                  calibration value
    {
      // read the value from the sensor:
      oneeighty = analogRead(sensorPin);

      // put oneeighty in EEPROM
      storeIntEEPROM(ONEEIGHTY_ADDRESS,
oneeighty);

      // Report the setting
      Serial.print("You set 180 degree to: ");
      Serial.println(oneeighty);
```



```

}
if(c == 's') // show the calibration values
{
    showCalibration();
}
if(c == 'r') // read the dial angle
{
    int val = analogRead(sensorPin);
    val = map(val,zero,oneeighty,0,180);

    // show the angle
    Serial.print("Angle: ");
    Serial.print(val);
    Serial.println(" degrees");
}
}
}

void showCalibration()
{
    // show calibration values on power up
    // get zero from EEPROM by multiplying by 4
    // to restore the 16 bit value (induces some
    // error)
    zero = retrieveIntEEPROM(ZERO_ADDRESS);
    Serial.print("zero = ");
    Serial.println(zero);

    // get oneeighty from EEPROM
    oneeighty = retrieveIntEEPROM
(OONEIGHTY_ADDRESS);
    Serial.print("oneeighty = ");
    Serial.println(oneeighty);
}

```

- ☐ Compile and upload the code.
- ☐ Open the serial monitor.
- ☐ Note that the start-up calibration values are random reads from the EEPROM.
- ☐ Set the pot angle to degrees and enter 'z' in the serial monitor.
- ☐ Set the pot angle to 180 degrees and enter 'o' in the serial monitor.
- ☐ Enter 's' to view the data stored in the EEPROM.
- ☐ Enter 'r' and note the angle displayed.
- ☐ Move the dial several times entering 'r' each time and verify that the results are similar to those in **Figure 10**.
- ☐ Do not dismantle this circuit since it will be reused in later labs.

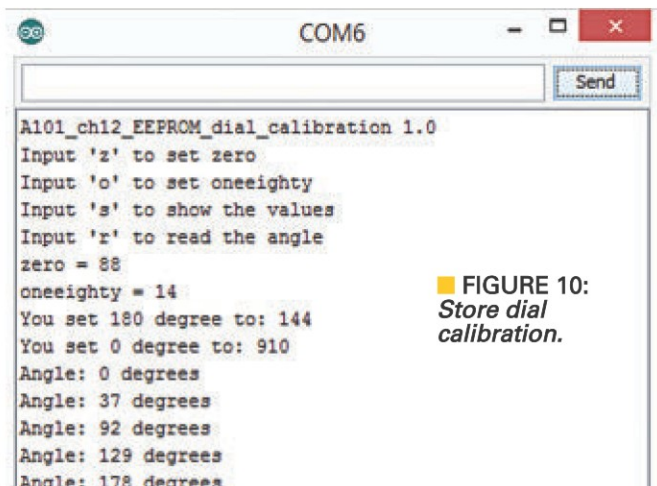


FIGURE 10:
Store dial calibration.

Lab 4: Store Dial Data

We will now use the potentiometer dial circuit to generate some test data that we will use in Lab 6 to display on the PC. The source code will allow us to slowly move the potentiometer from zero to 180 and back again several times over 10 seconds. We will record the sensor data 10 times a second, giving us 100 data points. If you move the potentiometer smoothly, you should get an approximation of a sine wave in this data set. This lab reuses the code from Lab 3, adding the steps to read the dial for 10 seconds and send the data to the PC. Since all the code is the same except for these two things, only the newly added code will be shown.

Parts required:

- 1 Arduino
- 1 USB cable
- 1 Arduino proto shield and jumper wires
- 1 Potentiometer
- 1 100 Ω resistor
- 1 Pot dial angle image

Estimated time for this lab: 15 minutes

Check off when complete:

- ☐ We will reuse the circuit we built for Lab 3.
- ☐ You can either modify the A101_ch12_EEPROM_dial_calibration.ino from Lab 3 as will be shown below or you can load the full source code by opening the A101_ch12_generate_dial_EEPROM_data.ino program.
- ☐ To modify the A101_ch12_EEPROM_dial_calibration.ino code, we first add this to the top:

```

#define DATA_ADDRESS 4 // start address after
the calibration values

```

In the setup() function we add:

```

Serial.println("Input 'g' begin reading 100
values for angles");
Serial.println("Input 'p' send the data to the
PC");

```

In the loop() function in the commands list we add:

```

if(c == 'g')
    // generate and save the dial angle data
{
    int temp = DATA_ADDRESS;
    // load starting address
    Serial.println("Starting to read the
dial");
    delay(500);
    // wait 1/2 a second to get started
    for(int i = 0; i < 100; i++)
    {
        val = analogRead(sensorPin);
        val = map(val,zero,oneeighty,0,180);
        EEPROM.write(temp++,val);
        // store data, increment the address
        delay(100);
    }
}

```


- ❑ Compile and open the serial monitor.
- ❑ Send the `g` command and move the dial from zero to 180 degrees several times over 10 seconds.
- ❑ Send the `p` command.
- ❑ Verify that you get results similar to those shown in **Figure 11**.
- ❑ Now, highlight and copy the data. What? It didn't work – bummer! You can't copy data from the Arduino serial monitor. So, let's use a serial monitor type program you can copy data from.

Lab 5: Send the Dial Data to the PC

As we just learned, the Arduino serial monitor doesn't let us copy data. We will now learn to use a serial terminal, PuTTY.exe, that will let us not just copy data, but load it to a log file on our PC.

Parts required:

- 1 Arduino
- 1 USB cable
- 1 PC

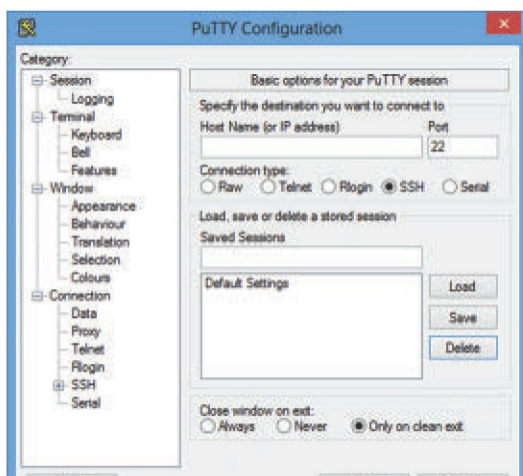
Estimated time for this lab: 30 minutes

Check off when complete:

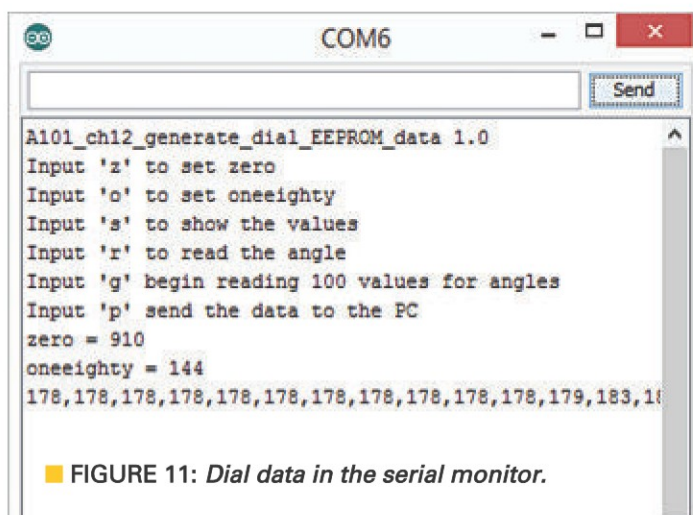
- ❑ Download PuTTY.exe from:
www.chiark.greenend.org.uk/~sgtatham/putty/download.html



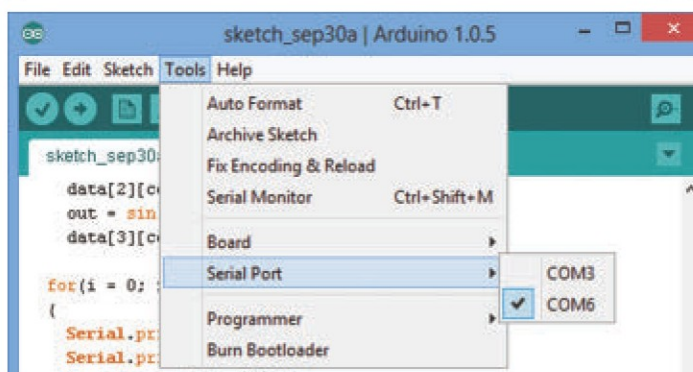
■ **FIGURE 12:**
PuTTY desktop icon.



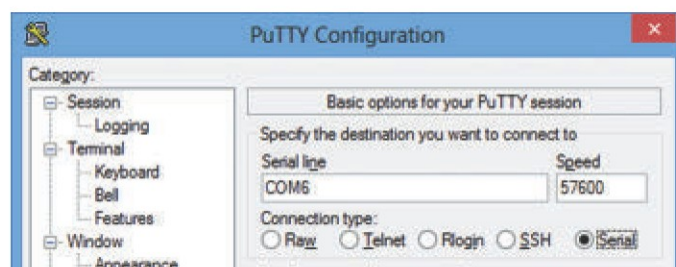
■ **FIGURE 13:**
PuTTY open window.



■ **FIGURE 11:** *Dial data in the serial monitor.*

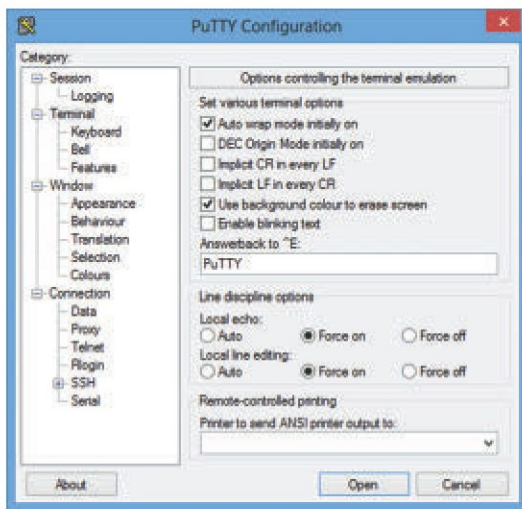


■ FIGURE 14: *Arduino serial port.*

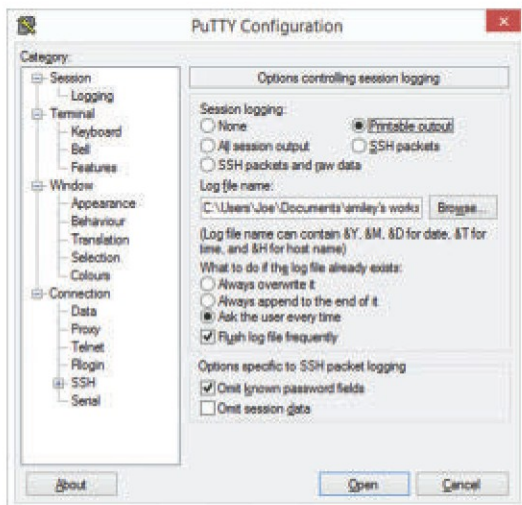


■ FIGURE 15: *PuTTY* serial line.

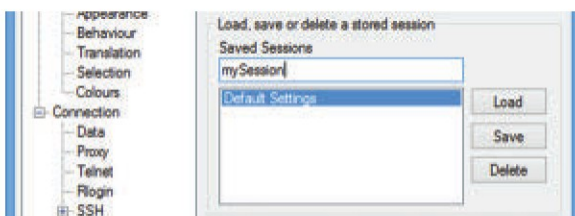
- ❑ Drag the PuTTY.exe file from your download directory to your desktop, leaving the icon shown in **Figure 12**.
- ❑ Open PuTTY as shown in **Figure 13**.
- ❑ Click on the 'Connection type:' radio button for serial, then enter the COM# for your Arduino. You can find this number by opening the Arduino menu Tools/Serial port as shown in **Figure 14**.
- ❑ Enter this COM# in the Serial line text box and enter the baud rate in the Speed box as shown in **Figure 15**.
- ❑ Click on the Terminal Category and in the 'Line discipline options,' click the 'Force on' radio buttons for both 'Local echo' and 'Local line editing' as shown in **Figure 16**.
- ❑ Create a file to save our logged data. Open the



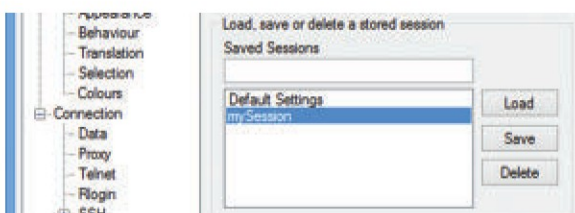
■ FIGURE 16: PuTTY terminal options.



■ FIGURE 17: PuTTY save log file.



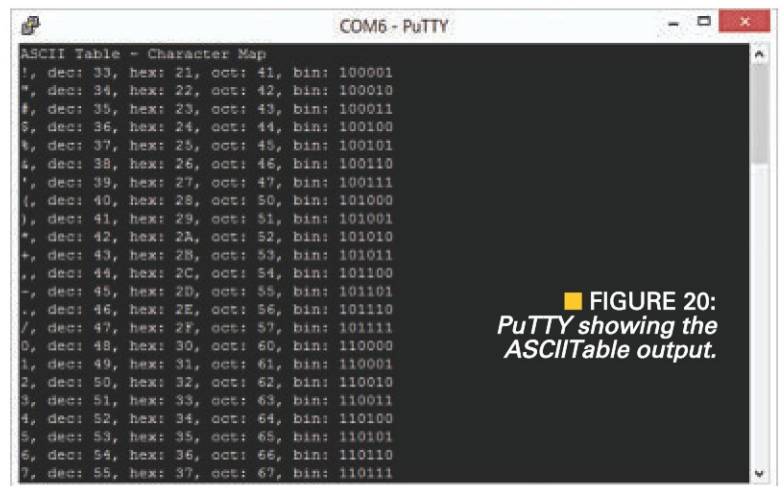
■ FIGURE 18: PuTTY store session.



■ FIGURE 19: PuTTY load session.

Sessions Logging window and click the 'Printable output' radio item. Next, we will create the file name dial_data.csv (more on the name in a moment) by clicking the browse button, navigating to the directory we want to use, then entering the file name as shown in **Figure 17**. We could now use PuTTY, but let's first save all the settings we just input so we can recall them the next time we want to use PuTTY.

- ❑ Return to the Category Session and save the session settings by typing in a session name (I'll use mySession) in the 'Saved Sessions' text box, then clicking the 'Save' button as shown in **Figure 18**.
- ❑ This will store the session as in **Figure 19**. The next time you open PuTTY, you'll have the option of clicking on the mySession item and the 'Load' button to have PuTTY load your custom settings.
- ❑ Now, we are ready to use PuTTY to log a session from the Arduino.
- ❑ In the Arduino IDE, open the Examples/Communication /ASCIITable file.
- ❑ Compile and run the program.
- ❑ Open a terminal session in PuTTY.
- ❑ Verify that you get output similar to that shown in **Figure 20**.



■ FIGURE 20: PuTTY showing the ASCII Table output.

Lab 6: Show Graph Data on a PC

We will now send the 100 bytes of dial data that we recorded in EEPROM to the PC using PuTTY. Then, we will edit the dial_data.csv log file in Notepad to remove the extraneous text that precedes the data. Finally, we will show the data in Google Chart.

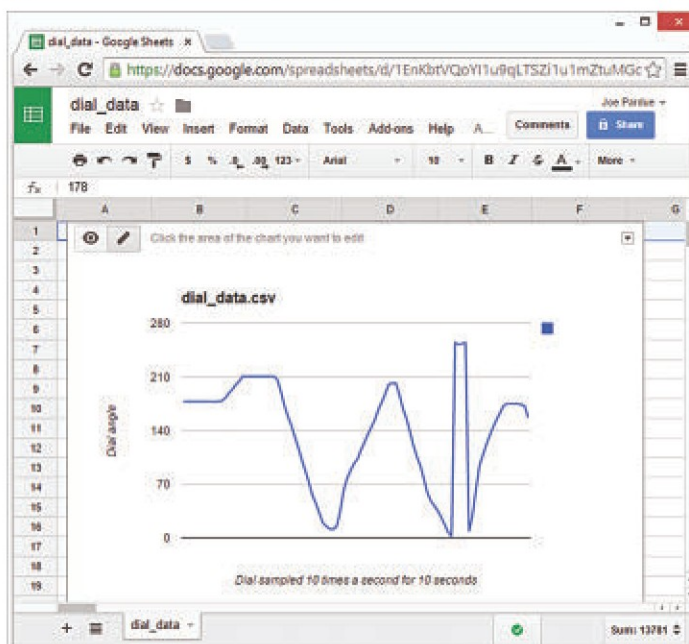
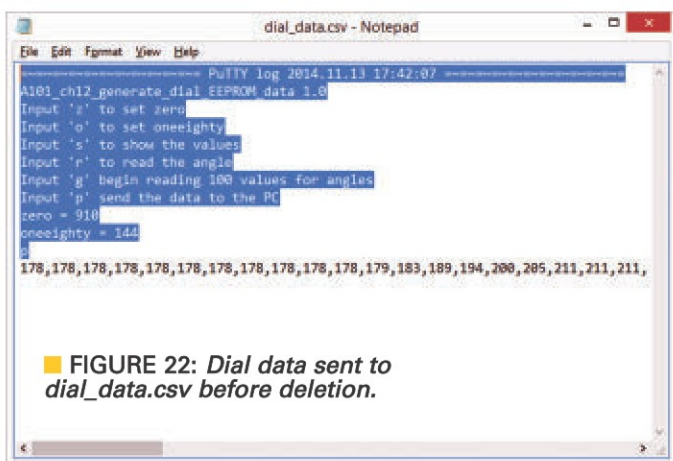
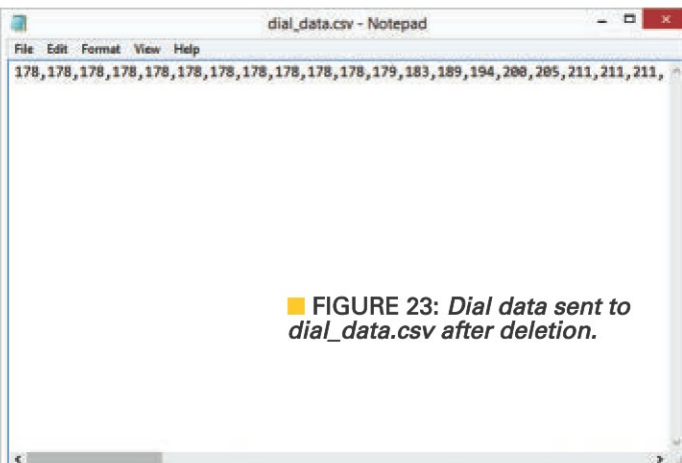
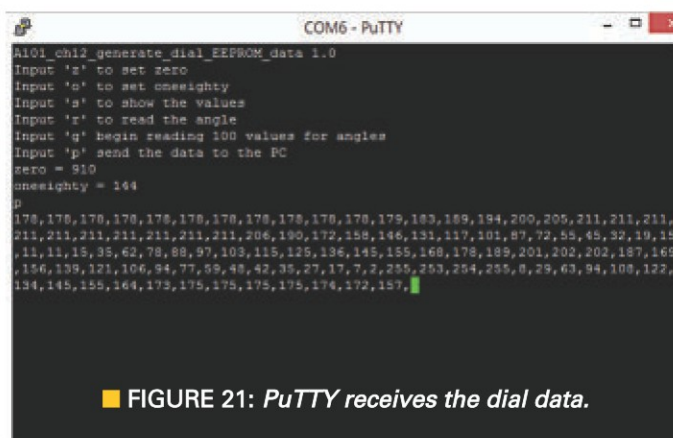
Parts required:

- 1 Arduino
- 1 USB cable
- 1 PC

Estimated time for this lab: 30 minutes

Check off when complete:

- ❑ On the PC, open a new instance of PuTTY.
- ❑ In the Arduino, run the A101_ch12_generate_dial_EEPROM_data program.
- ❑ In PuTTY, send the p command to the Arduino.



- ❑ Verify that you receive data similar to that shown in **Figure 21**.
- ❑ Open the dial_data.csv file in NotePad. Highlight the text from the Arduino that precedes that dial data. This should appear similar to **Figure 22**.
- ❑ Erase that highlighted data which should leave only the dial data as shown in **Figure 23**.
- ❑ Save the edited dial_data.csv.
- ❑ You will need to create an account for Google Drive. Since Google seems to change how this is done on occasion, we will not go into the details here. If the instructions are insufficient, then search Google for further information and play with the charting feature. It can be obtuse, but Google is very good at making things simple, so you will figure this out.
- ❑ In your file explorer, click on dial_data.csv and then drag and drop it onto your Drive page. It will then be uploaded to your Google Drive.
- ❑ In your Google Drive, click on CREATE and then click on spreadsheet.
- ❑ In the spreadsheet, click on File/Import and select the dial_data.csv file from your Google Drive.
- ❑ Click on the 'Insert Chart' icon on the spreadsheet toolbar.
- ❑ In the Chart Editor, click on the 'Switch rows/columns,'

then select a line chart and add text for the title and axes so the chart appears as in **Figure 24**.

- ❑ Examine **Figure 24** and see if you can notice a problem with the data.
- ❑ The dial goes from zero to 180, but the chart values exceed 180. What does this indicate?
- ❑ Probably, I swished the dial too far to the right exceeding the 180 mark, but nonetheless, this does give an example of how you can display a block of data and notice things that you might not have noticed in the raw data.

Next month, we will finish Arduino 101 by applying all that we have learned so far to build a battery-powered data logger that will measure temperature and light levels.

Serial I/O Data Interfaces: Part 1

Data is now transferred serially in most applications. However, there are lots of ways.

The two basic ways to transfer digital data from one place to another are parallel and serial.

Parallel data transfers send all bits of a word at once with one bit per wire. To send a byte, for example, you need eight wires or conductors in a cable (plus ground). To transfer that byte serially, all you need is one wire (plus ground). Each bit is sent for a fixed time period, one at a time in sequence. Today, both methods are still used, but in most applications a serial interface is implemented.

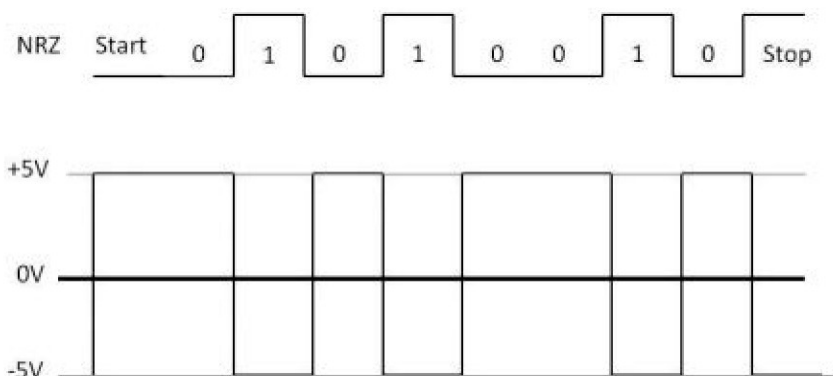
Parallel transfers are generally faster, but are more expensive as they require more hardware and wires. Furthermore, as parallel wires get longer, the inter-wire capacitance and the inductance of the wires greatly impacts data rates, so shorter connections are necessary to get any speed. Parallel connections run out of steam at about 100 Mb/s, and that can be achieved only over a foot or so. Serial is less expensive, simpler, and can achieve high speeds over longer distances.

There are literally dozens of different serial interfaces in use. Virtually all electronic devices today have a serial interface port of some kind. It is useful to know some of the basic details of these interfaces that you may use every day. This first article covers the slower interfaces such as CAN, I²C, RS-232, RS-485, and SPI. The second part of the article will cover USB, HDMI, PCIe, and a few others.

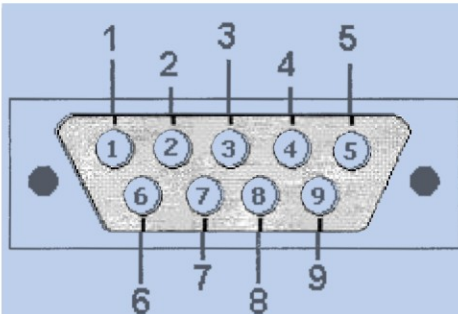
RS-232

The oldest serial interface is designated as RS-232 and was initially used in teletypewriters, computer video terminals, and modems. Later, it became a popular peripheral interface for PCs. Remember the PC serial port interface for printers and other devices? While PCs no longer have RS-232 serial ports, this interface is still around and found mostly in industrial equipment like programmable logic controllers (PLCs), robots, industrial computers, and some wireless gear. It is cheap and reliable.

The RS-232 interface transmits data seven or eight bits in sequence from LSB to MSB (see **Figure 1**). The standard NRZ (non-return to zero) binary data gets sent as a bipolar logic signal, where a binary 1 is some voltage between -3 and -25 volts and a binary 0 is a voltage level between +3 and +25 volts. Both +5 and +12 volt levels are common. Note that



■ **FIGURE 1.** Asynchronous data transmission with the RS-232 interface converts the normal NRZ (upper signal) into a bipolar RS-232 format shown in the lower waveform. Start and stop bits are added. The ± 5 V levels are typical.



■ **FIGURE 2.** The most commonly used connector for the RS-232 interface uses nine pins and is called a DB9 or DE9. Usually, only the RD and TD data lines are utilized, but other control lines are adopted for some protocols.

Pin Names		
1	DCD	Data carrier detect
2	RD	Receive data
3	TD	Transmit data
4	DTR	Data terminal ready
5	GND	Signal ground
6	DSR	Data set ready
7	RTS	Request to send
8	CTS	Clear to send
9	RI	Ring indicator

there is a start bit (logic 0) signaling the beginning of a word transfer and a stop bit (logic 1) that determines the end of the word. A parity bit is sometimes added between the end of the word and the stop bit for error detection.

This word format is implemented by a circuit called a universal asynchronous receiver transmitter, or UART. It may be a single IC or integrated on another chip such as a microcontroller. It handles all of the parallel-to-serial and serial-to-parallel data conversions, as well as adding and removing the start, stop, and parity bits. It also sets the speed.

As for the speed or data rate (R), it is simply the reciprocal of the time for one bit (t):

$$R = 1/t$$

As an example, for a bit time of 104.167 microseconds, the data rate is 9600 bits per second (bps or b/s).

Knowing the data rate, you can find the bit time:

$$t = 1/R$$

If the data rate is 1 Mbps, the bit time is one microsecond.

The RS-232 interface defines multiple data rates depending on the application. Some common standard rates are 9600 bps, 19.2 kbps, and 115.2 kbps. The upper limit is really determined by the cable length. Long cables have lots of capacitance, so rounding off the leading edges of the binary pulses minimizes the data speed. Cable lengths are generally

less than 50 feet.

The RS-232 standard also specifies a connector. The early versions used a 25-pin connector, but today a nine-pin connector is the most common (refer to **Figure 2**). Early applications used several of the control lines to implement communication protocols, but today mostly only the transmit (TD) and receive (RD) lines are used.

I²C

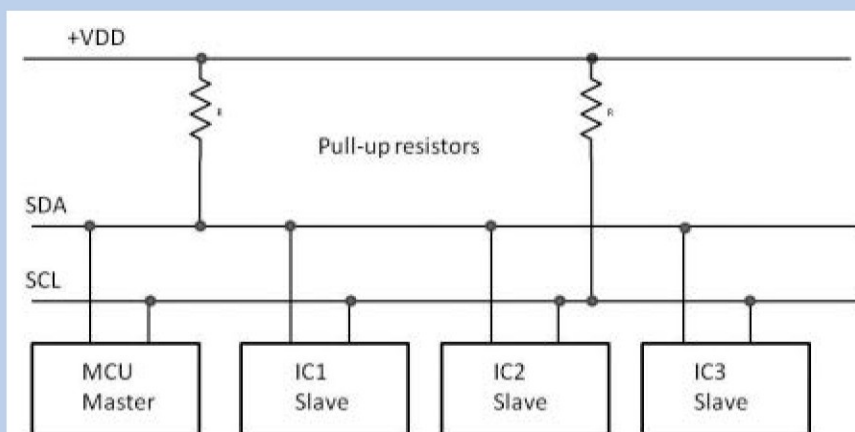
I²C, or the Inter-Interconnect bus is an interface developed by Philips (now NXP Semiconductor) to connect one complex IC to another on a printed circuit board (PCB). The transmission medium is just the copper pattern on the PCB. It can also use a short cable to connect two PCBs or other devices.

The physical arrangement consists of a master control IC like a

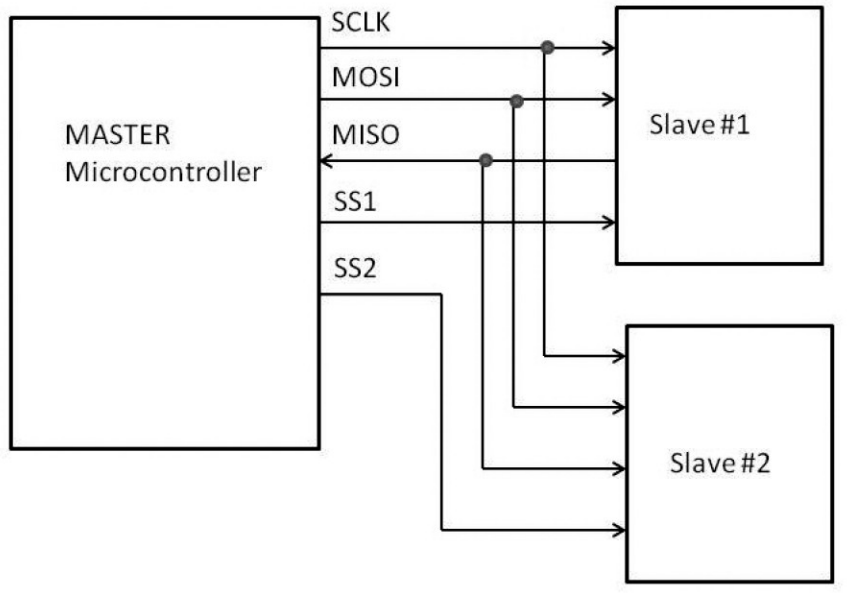
microcontroller and one or more slave devices that receive the data or send some of its own back to the master (**Figure 3**). Note that there is a data line (SDA), a clock line (SCL), and a ground wire forming a bus. Up to about 30 slave devices can be accommodated if the bus connections are not too long. Bus length is usually less than one foot on a PCB or no more than several feet for a short cable. Bus length limits data rate.

Data is sent by way of a frame or packet that contains a start bit, a seven- or 10-bit slave address, a read/write bit followed by the data bytes to be sent with each followed by an acknowledge bit (Ack), and finally a stop bit. The basic data rate is 100 kbps, but higher speeds such as 400 kbps, 1 Mbps, and 3.4 Mbps can be used if the bus capacitance is not too great.

There are several variations of the I²C bus. A common one is the system management bus (SMB) developed by Intel. It is used to communicate between chips on PC motherboards. A variation of the SMB is the power management bus (PMB) that is used for controlling and monitoring power supplies and ICs like regulators or DC-DC converters. Another version is the two-wire interface (TWI) used by Atmel.



■ **FIGURE 3.** The popular I²C interface uses a master and one or more slave devices. External pull-up resistors are used on the MOSFET drivers inside each chip. The typical application is intercommunications between ICs.



■ **FIGURE 4.** Another popular chip-to-chip interface is SPI. It is faster than I²C and uses a few more lines to link a master controller to one or more slaves. Each slave is enabled with a slave select (SS) line.

SPI

The serial peripheral interface (SPI) is similar to the I²C bus since its primary application is interconnecting chips on a PCB or linking two PCBs over a short cable. The primary difference is that SPI is significantly faster. SPI is integrated into many fast embedded controllers, as well as

memory devices like Flash, I/O chips, DACs, ADCs, FPGAs, and other fast peripherals.

Data rates are usually 1 Mbps or more, up to about 50 Mbps. Range is no more than a few feet, and usually only inches on a PCB.

Figure 4 shows the usual connection. Master devices – typically, the microcontroller – select a slave with a slave select (SS) line

and initiate a read or write operation. Note the two data lines: master out slave in (MOSI) and master in slave out (MISO). A clock line times the synchronous transfer of data by bytes or longer words. There are no start or stop bits. Multiple slave units can be accommodated. A slave select line is needed for each.

CAN

The controller area network (CAN) bus is especially designed for automotive applications. It is used to transfer data between multiple electronic control units (ECUs) in a vehicle such as sensors, actuators, and other devices to control the transmission, engine, suspension, brakes, or other systems. It is also used in some industrial control applications.

CAN is especially reliable as it uses shielded twisted pair cable that can be many meters long. The differential pair cable, along with the shield, effectively reduce the noise level in harsh environments, thereby producing fewer bit errors. Data rates depend on usage, but can range from 10 kbps up to 1 Mbps.

The wire bus is differential with two lines (CANL and CANH) for high and low signals. Up to 127 nodes



PBP3

PICBASIC PRO™ Compiler
BASIC compiler for Microchip PIC® microcontrollers

FREE
Student - ~~\$49.95~~
Silver Edition - ~~\$119.95~~
Gold Edition - ~~\$269.95~~

Download **FREE** trial now

Simple, Reliable, Affordable

WWW.PBP3.COM

PICBASIC and PICBASIC PRO are trademarks of Microchip Technology Inc. in the USA and other countries. PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.

may be paralleled along the bus. The bus can run up to 1,000 meters but that limits the data rate to no more than 40 kbps at that range. Each node on the bus is a transceiver that may send or receive data. Nodes compete for bus usage with a contention process that gives the bus to the node with the highest priority. Data is then sent byte by byte with asynchronous start and stop bits.

There are several variations of CAN, including CAN 2.0A and CAN 2.0B. Others are CANopen used in industrial applications, DeviceNet used in factory automation, and J1939 – a Society of Automotive Engineers (SAE) variant for trucks and buses.

CAN may be integrated into microcontrollers or other ICs. However, separate CAN transceivers and controllers are also available.

RS-485

Another popular serial standard is RS-485. Like CAN, it uses differential shielded or unshielded twisted pair cable to form a bus. The differential configuration helps cancel noise over long distances. Cable length can be up to 4,000 feet, but it does limit the data rate to about 100

kbps. Data rates to 10 Mbps can be achieved at up to 40 feet. Depending on the type of line driver and bus length, speeds up to about 35 Mbps can be achieved.

The bus can have up to 32 nodes on it. Each node is a transceiver that may send or receive. Separate IC transceiver chips are commonly used.

As for applications, it is variable. Mostly, RS-485 is found in industrial applications but it also shows up in building automation, video surveillance, and point of sale terminals.

There is no fixed transmission protocol, so any solution can be used. Transmission is usually asynchronous byte by byte using start and stop bits.

OTHER INTERFACES

As I said earlier, there are dozens of other serial I/O interfaces. Just a few are 1-Wire, BitBus, FlexRay, IOLink, I²S, LIN, Microwire, MIDI, OBD, and X10. Most were developed for specific applications and many are similar in operation. Some of the faster interfaces are more widely used. I will cover the most popular of these like USB, Ethernet, and PCIe in Part 2. **NV**

Feedback Motion Control

The Old Way

- 1) Build robot
- 2) Guess PID coefficients
- 3) Test
 - 3a) Express disappointment
 - 3b) Search Internet, modify PID values
 - 3c) Read book, modify PID coefficients again
 - 3d) Decide performance is good enough
 - 3e) Realize it isn't
 - 3f) See if anyone just sells a giant servo
 - 3g) Express disappointment
 - 3h) Re-guess PID coefficients
 - 3i) Switch processor
 - 3j) Dust off old Differential Equations book
 - 3k) Remember why the book was so dusty
 - 3l) Calculate new, wildly different PID coefficients
 - 3m) Invent new, wildly different swear words
 - 3n) Research fuzzy logic
 - 3o) Now it is certainly not working in uncertain ways
 - 3p) Pull hair
 - 3q) Switch controller
 - 3r) Re-guess PID coefficients
 - 3s) Switch programming language
 - 3t) Start a new project that doesn't need feedback control
 - 3u) See parts in box. Feel guilty. Go back to old project
 - 3v) Start testing every possible combination of PID coefficients
 - 3w) Apply eye drops to red, bleary, sleep-deprived eyes
 - 3x) Wait. It's working!
 - 3y) Decide not to do any more projects that require control systems
 - 3z) Wonder why someone doesn't just make a thing that tunes itself

The Kangaroo x2 Way

- 1) Build robot
- 2) Press Autotune
- 3) Get a snack

Kangaroo x2 adds self-tuning feedback to SyRen and Sabertooth motor drivers.



\$24.99

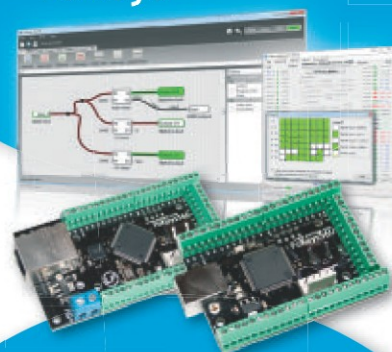
DimensionEngineering

www.dimensionengineering.com/kangaroo

Imagine this...

- a small and inexpensive
- USB or ethernet device
- with a rich set of features
- including PLC functionality and up to 8-axis stepper motor controller
- all accessible via free .NET, ActiveX or C++ library
- cross-platform
- configurable with free software

PoKeys



Or this...

- all in one: Oscilloscope, Data Recorder, Logic analyzer, Analog and digital signal generator
- smallest USB 2.0 portable 1MS/s oscilloscope
- data acquisition of analog and digital signals
- data recording
- export to CSV, XLS, PDF and HTML
- simple usage of advanced features
- examples for C++, VB, Delphi and LabView
- free software and updates

PoScope Mega1+



Polabs
Visit us at
www.poscope.com



ELECTRONET

CORIDIUM
Floating point **BASIC** for
ARM controllers from \$5.00
www.coridium.us

GET THE NUTS & VOLTS DISCOUNT!
Mention or enter coupon code **NVRMZ142**
and receive 10% off your order!
ramsey
www.ramseykits.com
AM/FM Broadcasters • Hobby Kits
Learning Kits • Test Equipment
...AND LOTS OF NEAT STUFF!

USB Add USB to your next project--
It's easier than you might think! **DLP Design**
USB-FIFO • USB-UART • USB/Microcontroller Boards
RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!
www.dlpdesign.com

INVEST in your BOT!
HITEC
12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecusa.com

HOBBY ENGINEERING
Kits, Parts and Supplies
www.HobbyEngineering.com

MYRO www.myropcb.com
PCB, PCBA and More
LOW cost with HIGH Quality
1-888-PCB-MYRO

Micro BGA SMT Adapter
0.5mm Pitch
Ironwood ELECTRONICS
www.ironwoodelectronics.com

THE ORIGINAL SINCE 1994
PCB-POOL
Beta LAYOUT
• Low Cost PCB prototypes
• Free laser SMT stencil with all Proto orders
WWW.PCB-POOL.COM

superbrightleds.com
COMPONENT LEDs • LED BULBS • LED ACCENT LIGHTS

ALL ELECTRONICS CORPORATION
Electronic Parts and Supplies.
www.allelectronics.com
Free 96 page catalog 1-800-826-5432

For the ElectroNet
online, go to
www.nutsvolts.com
click **Electro-Net**

**Did You Know Preferred
Subscribers get access to all the
digital back issues of
Nuts & Volts for free?**
Call for details
1-877-525-2539

www.servomagazine.com

MOBILE APP NOW AVAILABLE!
SERVO
Download NOW On Your
Favorite Mobile Device!
FOR ROBOT BUILDERS
iOS • ANDROID • KINDLE FIRE



A FASTER WAY TO ACHIEVE BLUETOOTH® SMART CONTROL OF AN EMBEDDED DEVICE IS COMING!

JOIN THE EVOLUTION

JANUARY 2015



www.anaren.com
800-411-6596
In Europe: 44-2392-232392

Anaren®
What'll we think of next?™

Atmosphere Takes Bluetooth Apps to New Heights

Editor's note: The following article/preview of Anaren's new AIR for Wiced/Atmosphere development portfolio is based on a 'beta' provided to *Nuts & Volts* as a courtesy. The official launch of the complete AIR for Wiced/Atmosphere solution and its various components (including the online Atmosphere development tool) will be at the CES show on January 6, 2015. Until then, interested parties can pre-register to be notified when the solutions "go live" at this web address: www.anaren.com/new-air-bluetooth-solution-coming-soon.

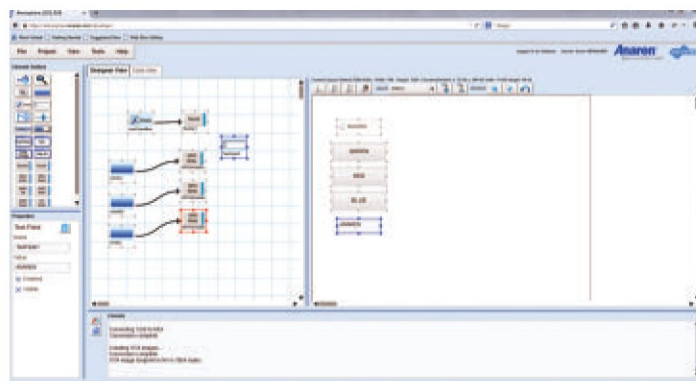
Anaren is doing for Bluetooth what FTDI has done for USB. In the days of the dinosaur, cavemen like me used code made of sticks and stones to realize a meager USB interface. At about the same time we discovered fire, the FTDI guys emerged from a cave with a brand new easy way of doing USB. Bluetooth is a very capable wireless communications method as was proven in my trio of Design Cycle PlayStation DualShock 3 columns (May, June, and July 2013). However, as I recall, the major hurdle was not the coding itself, but getting around to understanding the application of the inner layers of the Bluetooth protocol. I was recently joking around with some friends and predicted that in the near future all newborns would be issued an iPhone and a debit card at birth. You may be laughing, but Anaren has put some meat behind my prognostication. You are about to be introduced to something you thought would never happen in your lifetime. I am going to introduce you to the tools that will allow you to simultaneously code an embedded Bluetooth monitor/control application and its associated iPhone GUI app. The methods we are about to discuss and explore are made possible by a web-based tool that Anaren calls Atmosphere.

A Bluetooth Breath of Fresh Air

Atmosphere is a web-based design tool that works in conjunction with the Atmosphere programmer. As you can see in **Screenshot 1**, the Atmosphere development tool is an object-oriented Bluetooth development platform. As application objects from the Element Toolbox are added to the Designer View pane, associated code stubs are inserted into the Code View pane. Application object properties are managed in the Properties pane. Any graphical representations of the selected application objects are placed in the layout window, located at the far right of **Screenshot 1**. The placement of the graphics is under programmer control.

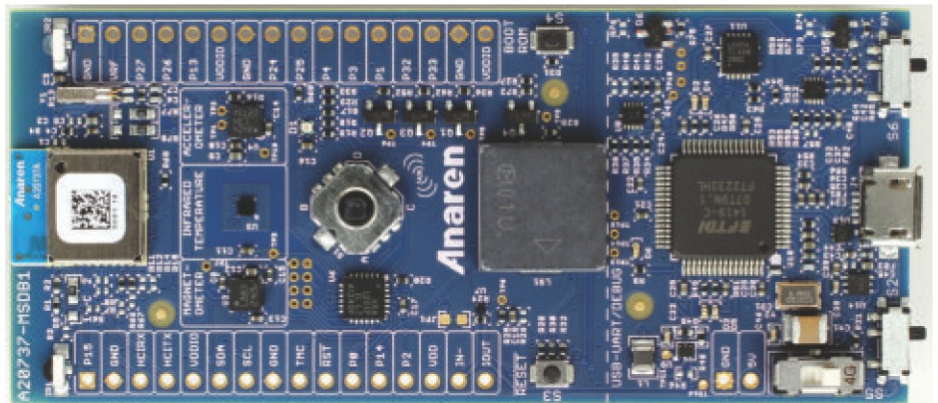
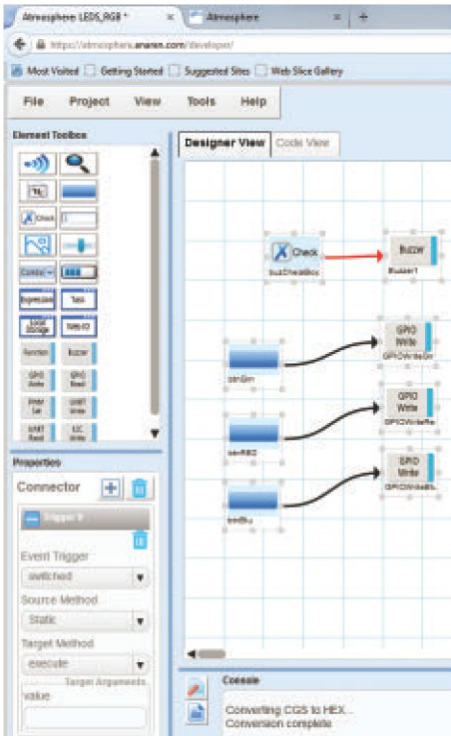
Clicking inside the application object widgets reveals a "connector" graphic, which is the "arrow-terminated" wire you see between some of the application objects in the Designer View. The connector highlighted in **Screenshot 2** makes things happen on the buzzer end when the checkbox is "switched." Likewise, the connectors positioned between the buttons and GPIO Write application objects are responsible for firing the I/O pins that toggle the illumination of the associated LEDs.

For grins and giggles, I've added a textbox to the design. Tapping buttons on the iPhone glass and making real world things happen should be priming that light bulb that's about to go off in your mind's eye. Take a quick look back at **Screenshot 1** and you will see that I've arranged the buttons, checkbox, and textbox in an eye-pleasing



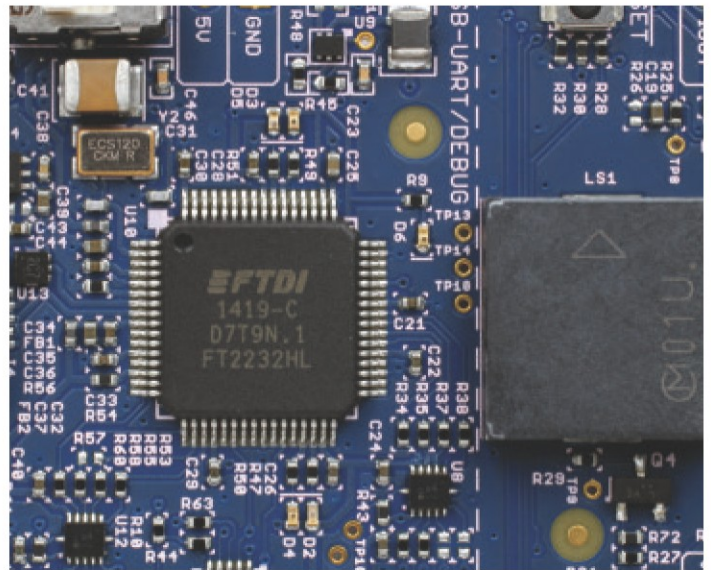
■ Screenshot 1. This is a view of the Atmosphere web window. The act of selecting objects from the Element Toolbox spawns behind-the-scenes graphics and code stub creation processes.

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2015_DesignCycle.



■ Photo 1. The Anaren Bluetooth development kit is built around the Anaren A20737A Bluetooth module. This puppy is loaded with sensors that are easily accessed via your smartphone.

■ Screenshot 2. The connector may seem insignificant. However, it plays a very important part in the behavior of the application.



■ Photo 2. The FTDI FT2232HL provides USB connectivity to the multi-sensor development kit.

symmetrical configuration. We are poised to kick off a Bluetooth mobile app that controls LEDs and a buzzer on an associated piece of radio-equipped hardware.

The Anaren Multi-Sensor Development Kit

Behold **Photo 1**. Everything hardware needed to interface sensors and the touch screen of a smartphone is shown in this overhead shot. Let's take a closer look at the component makeup of the Anaren multi-sensor development kit. On the USB side of the dotted line in **Photo 2**, we find an FTDI FT232HL. The FT232HL is a dual-interface USB-to-whatever IC, with "whatever" being bit-bang, JTAG, I²C, or SPI.

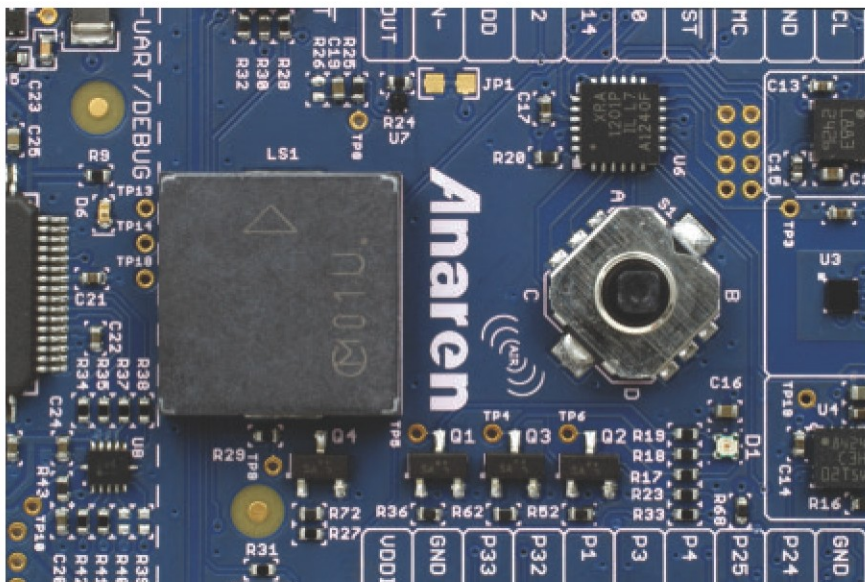
You can see the channel activity LED pairs directly above and below the FT2232HL. At the top pair center of **Photo 2** lies the resident step-down switching regulator (U9) which is based on a TI TPS62730. The resources of the crystal to the upper left of the FT2232HL (Y2) belong to the FT2232HL. U8 and U12 – which you see directly below the FT2232 – are buffers that perform a logical voltage translation between the FT2232HL and the A20737A I/O pins.

Note the obvious absence of a microcontroller. This implies that the “smarts” of the Bluetooth system resides within the radio module. The logical design location of the logic level converters (U8 and U12) tends to indicate that the data buck stops at the radio. Another clue that there may be a microcontroller under the bonnet of the radio is

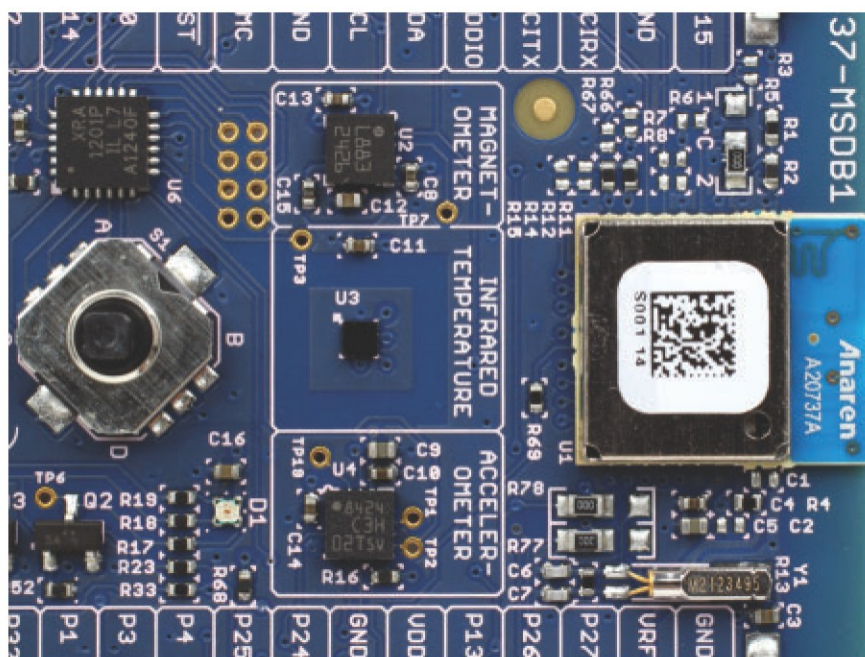
the silkscreen associated with the I/O pin header positions.

Directly across the dotted line lies what at first glance appears to be the monolith that the apes (and humans) were attracted to in *2001: a Space Odyssey*. Well, it's certainly not some sort of range extender or a ferrite bead (more like a ferrite block). A bit of web research reveals that the monolith you see prominently in **Photo 3** is actually a muRata buzzer. The trio of MOSFET drivers (Q1, Q2, Q3) along one side of the buzzer drives an integrated RGB (Red/Green/Blue) LCD module, which is pictured in the lower right corner (D1). MOSFET Q4 drives the buzzer.

Moving in the direction of the radio module, the next landmark we encounter is a digital joystick. The joystick is not really a "joystick" in the analog sense, but a series of four SPST switches (A, B, C, D) that are fired in the rotation of the "stick." The 24-pin QFN device nearest to



■ Photo 3. The buzzer is a real fooler. We are so used to seeing buzzers with a round housing with a center sound hole. The digital joystick you see in this shot is actually just a 4PST rotational switch (A, B, C, D) with an additional SPST switch in the push position.



■ Photo 4. The hardware in this shot is obvious to the most casual observer. Normally, there would be some kind of microcontroller included in the design to handle the interfacing to the sensors and the radio. There's a crystal in the mix (Y1), but still no sign of a microcontroller.

the joystick is an Exar XRA1201 I²C/SMBus-controlled I/O expander. The XRA1201 services the joystick switches and certain signals associated with the sensors.

Just beyond the joystick and XRA1201 I/O expander lies the sensor field shown in **Photo 4**. Ambient temperature or the temperature of the nearest substantial object is measured using a TI TMP006B. The TMP006B

measures the temperature of an object without the need to make contact with the object. This sensor uses a thermopile to absorb the infrared energy emitted from the object being measured, and uses the corresponding change in thermopile voltage to determine the object temperature. The TMP006B rides on the SMBus.

A Honeywell three-axis digital compass IC (MHC5883L) gives the multi-sensor development kit direction. The MHC5883L is an I²C device capable of delivering very accurate single-digit tolerance compass headings. If you're a serious Arduino head looking for some direction in life, you've seen this part before.

The sensor field is also equipped with an STMicro LIS3DH accelerometer IC. The LIS3DH can ride the I²C wires, as well as speak the SPI protocol. In this case, the LIS3DH differs from the other sensors in that its SPI portal is tapped as the communications conduit.

The radio module sitting at the tip of the spear is Anaren's new A20737A AIR module. The A20737C is FCC and IC certified and ETSI-compliant. The A20737A features Bluetooth Smart technology that incorporates the Broadcom BCM20737 SOC. The A20737A is tiny, measuring in at 11 x 13 x 2.5 mm.

Just as we suspected, the A20737A AIR module incorporates a hefty microcontroller, an EEPROM, a 24 MHz crystal, and the required RF matching and filtering for regulatory compliance. The A20737A AIR module operates in the global unlicensed 2.4 GHz ISM/SRD frequency band.

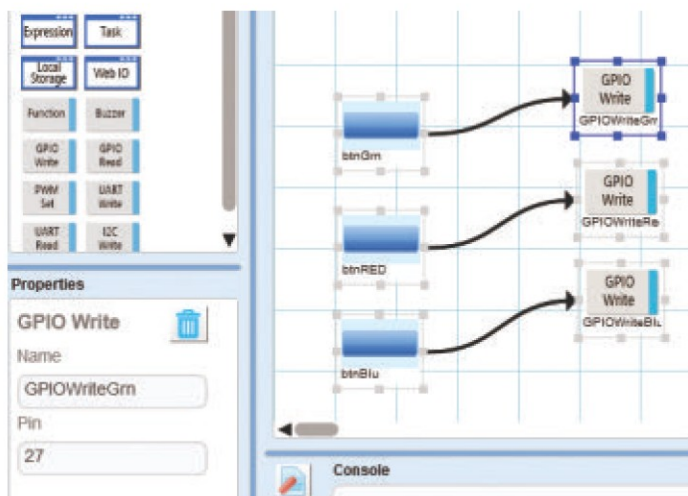
You can get a more detailed description of the multi-sensor development kit hardware by downloading the schematic diagram from the Anaren website. For now, all you really need to visualize is a standard MOSFET driving an LED by providing the LED a path to ground when the MOSFET gate is driven logically high.

Visualize that three times: once for the RED LED, again for the GREEN LED, and one more time for the BLUE LED. Recall that these LEDs are part of the tri-color LED package we saw in **Photo 3**. I've provided a schematic snippet to help you more easily draw this mental image. The actual MOSFET driver/LED module hardware is depicted in **Schematic 1**.

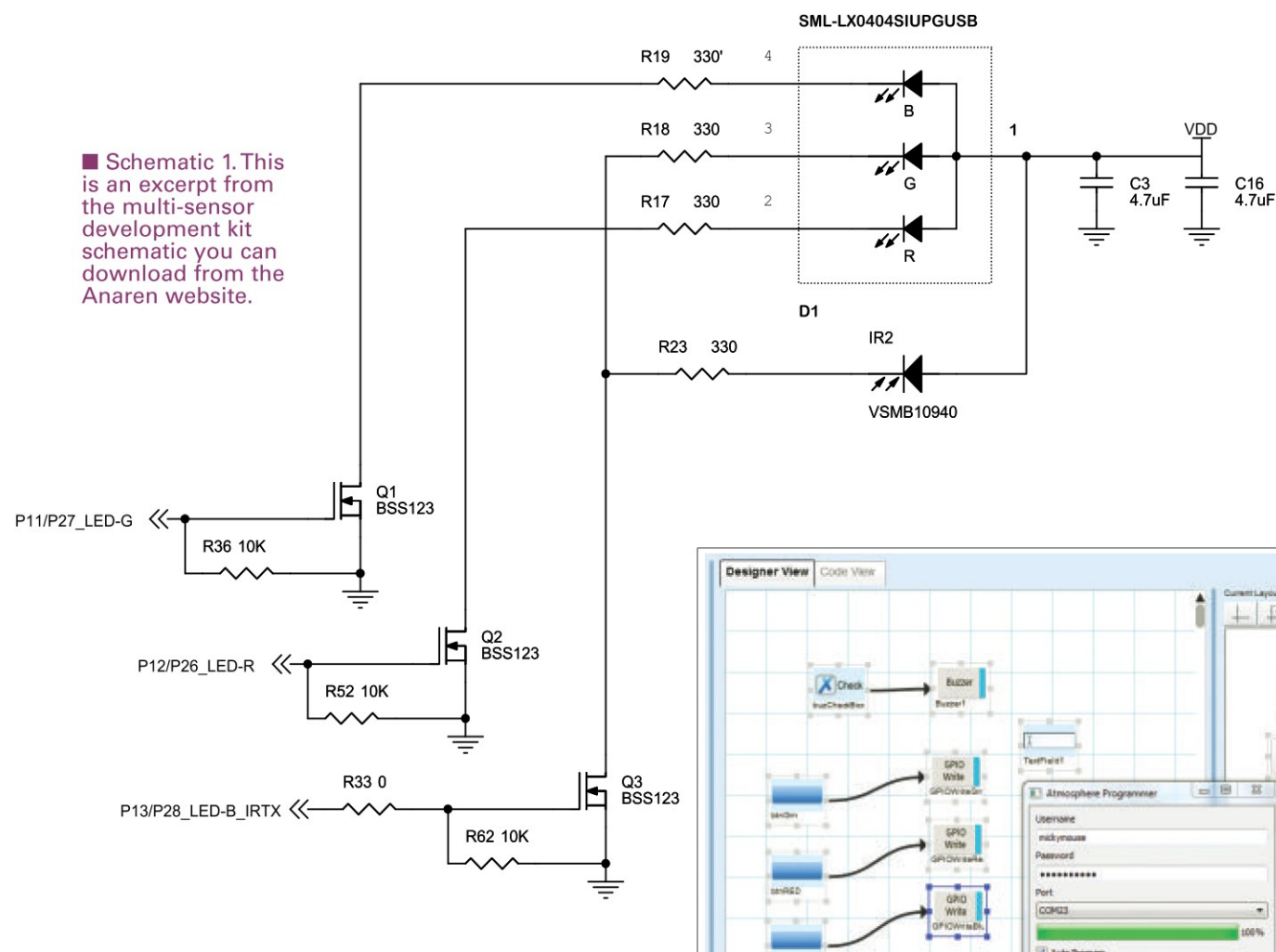
A Very Expensive LED Controller

Let's pick up where we left off in **Screenshot 2**. We are going to use the Atmosphere tools to build a simple application to control the tri-color LED module with an iPhone, which happens to be a model 5S. As you can see in **Screenshot 2**, I have assembled a set of button elements that are acting as logic triggers for the LEDs we want to control. I've also placed a checkbox in control of the multi-sensor development kit's buzzer.

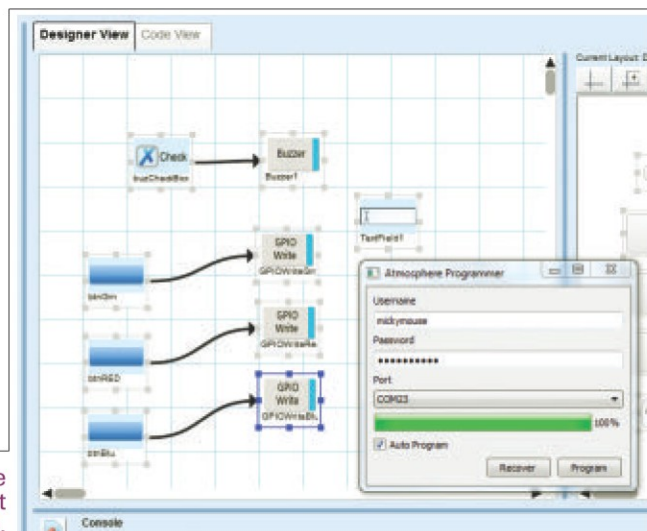
Drawing from the kit's schematic snippet, I determined that the gates of the MOSFETs that are driving the LED module are electrically attached to particular A20737A GPIO pins. As you can see, the green LED is controlled via GPIO pin 27. The red LED is under the thumb of GPIO pin 26, and the blue LED is switched by the MOSFET gate attached to GPIO pin 28. I assigned a logical GPIO element to the green LED in **Screenshot 3**.

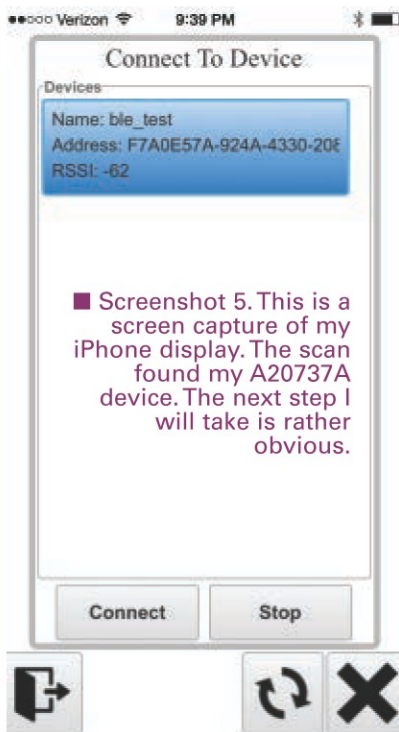


■ Screenshot 3. The Properties pane allows me to assign a logical button trigger and GPIO control element to physical pin 27 of the A20737A AIR module.



■ Screenshot 4. The Atmosphere programmer loads the application code into the multi-sensor development kit hardware via the FT2232HL and its USB interface.





■ Screenshot 5. This is a screen capture of my iPhone display. The scan found my A20737A device. The next step I will take is rather obvious.

As you might expect, I performed the same GPIO assignment operation to the rest of the LEDs. By the way, notice that the blue LED is also sharing a GPIO control pin with an IR emitter. You can see an IR emitter and detector (IR1, IR2) flanking the A20737A AIR module in **Photo 1**. I didn't mess with the iPhone layout I arranged in **Screenshot 1**.

So, at this point we can try to build our application. After making sure the application would build successfully, I fired up the

Atmosphere programmer. As you can see in **Screenshot 4**, I elected to allow the programmer to automatically program the multi-sensor kit's hardware upon a successful application build operation. According to **Screenshot 4**, everything went well including the application program process. So, we should be able to "dial up" the kit with my iPhone. With that, I kicked off the Atmosphere app on my phone and scanned for my development kit's A20737A. The results of my scan are shown in **Screenshot 5**.

Following the connect, I was presented with the display you see in **Screenshot 6**. This screen capture should be familiar. Tapping the BUZZER checkbox woke up the multi-sensor kit's buzzer. A second tap silenced the buzzer. I was able to control the LEDs in a similar manner. A tap illuminates and a tap extinguishes. By the way, did you notice that I didn't write a single line of code?

The Atmosphere Code View

However, something or somebody did plink out some code. Actually, the code was being generated as I was moving elements around and exposing their connectors. Here's the initialization code which is generated according to the elements that are included in the design:

```
void setup() {
    AIR_GPIO_Init();
    AIR_UART_Init();
    AIR_I2C_Init();
    AIR_SPI_Init();
    BUZZER_Init();
}
```

Every subsystem that was initialized supports a piece of development kit hardware. Let's take a look at life for the buzzer after initialization:

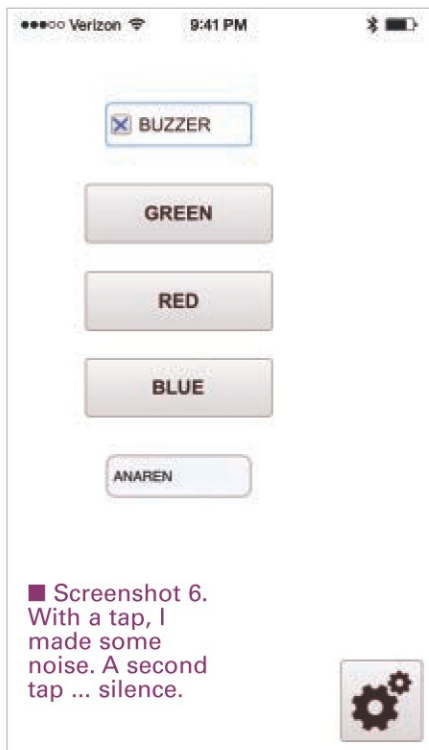
```
char* Buzzer1(char* data, int length) {
    if(data[0] == '1' || data[0] == 't')
    {
        BUZZER_Start(%(note)s);
    }
    else if(data[0] == '0' || data[0] == 'f')
    {
        BUZZER_Stop();
    }
    else
    {
        static bool toggle = false;
        toggle = !toggle;
        if(toggle)
        {
            BUZZER_Start(%(note)s);
        }
        else
        {
            BUZZER_Stop();
        }
    }
    return NULL;
}
```

The buzzer can be controlled using values. An ASCII "1" or a "t" will turn the buzzer's MOSFET driver on. Conversely, an ASCII "0" or "f" will kill the sound. If no values are involved, the buzzer is controlled according to the state of *toggle*.

If *toggle* is TRUE, the buzzer is activated. A Boolean FALSE assigned to *toggle* will terminate the buzzer MOSFET's gate drive. Note that the Boolean value of *toggle* is swapped every time the code is called. Thus, consecutive taps on the buzzer control will alternately produce sound and silence.

The same logic that applies to the buzzer handler code is also applied to the GPIO elements. Here's what the blue LED driver code looks like:

```
char* GPIOWriteBlu(char* data, int length) {
    if(length > 0)
    {
        if(data[0] == '1' || data[0] == 't')
        {
            AIR_GPIO_Write(%(pin)s, 1);
        }
        else if(data[0] == '0' || data[0] == 'f')
        {
            AIR_GPIO_Write(%(pin)s, 0);
        }
    }
}
```

```

    else
    {
        bool toggle =
            !AIR_GPIO_Read(%(pin)s);

        AIR_GPIO_Write(%(pin)s,
            toggle);
    }
    else
    {
        bool toggle =
            !AIR_GPIO_Read(%(pin)s);

        AIR_GPIO_Write(%(pin)s, toggle);
    }

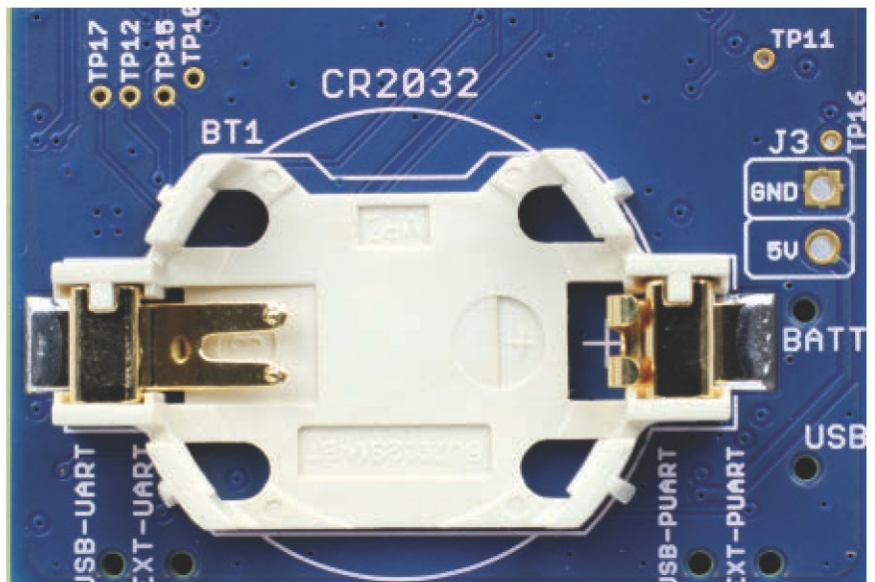
    return NULL;
}

```

There is nothing significantly different here in the logic area. Instead of brute force GPIO pin manipulation, the GPIO elements are steered from an API (Application Program Interface) call (*AIR_GPIO_Write*). The red and green LED driver code is identical with the exception of the target GPIO pin.

Free Fall

Up to this point, we've been depending on the multi-sensor development kit's USB connection for power and intelligence. Your iPhone doesn't run around with a cable attached. The device your iPhone is destined to control may not have the luxury of a nurturing cable either. So, it seems that we may need a well-placed battery.



■ Photo 5. I was musing about newborns and predicted they would be issued iPhones and debit cards. I forgot to add a very important item to the newborn's kit: the ubiquitous CR2032 coin cell.

That thought must have run through the Anaren folks' minds, as well. If you turn the development kit over on its back, you will find a coin cell holder. As you can see in **Photo 5**, the coin cell holder is home to the popular CR2032. To employ the services of the CR2032, we need only to populate the coin cell holder, load our Atmosphere-generated code, and throw slide switch S5 to the BATT position (which is shown in silkscreen in **Photo 5**). You can locate S5 which is on the opposite side of the board using **Screenshot 1**.

The Possibilities

If you take a look at the selection of Atmosphere Elements, the A20737A AIR module can very easily be coaxed to communicate with components that speak SPI, I²C, and RS-232. We can even communicate with devices or components on a direct logical pin-to-pin level. In addition to the ability to access sensors, other microcontrollers, and utility ICs directly from the radio module, we also have immediate access to the Bluetooth RF channel on the other side.

With the advent of Atmosphere and the A20737A AIR module, it is now possible to visualize, build, load, and test a smartphone-based Bluetooth control application without spending thousands of dollars and investing hundreds of man hours of development time. By the way, you don't need an iPhone to utilize Anaren's Atmosphere development tool environment. Atmosphere also works with Android. **NV**

For complete product details, visit our webstore!!

The Nuts & Volts **WEBSTORE**

GREAT FOR DIYers!

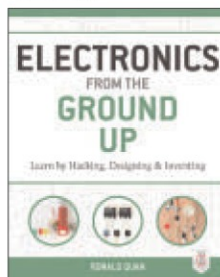
Electronics from the Ground Up: Learn by Hacking, Designing, and Inventing

by
Ronald Quan

Are you fascinated by the power of even the smallest electronic device? Electronics from the Ground Up guides you through step-by-step experiments that reveal how electronic circuits function so you can advance your skills and design custom circuits. You'll work with a range of circuits and signals related to optical emitters and receivers, audio, oscillators, and video.

Paper back 544 pages.

\$30.00



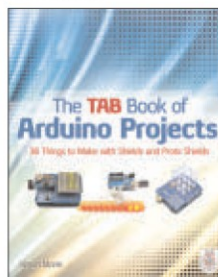
The TAB Book of Arduino Projects

by Simon Monk

The ultimate collection of DIY Arduino projects!

In this easy-to-follow book, electronics guru Simon Monk shows you how to create a wide variety of fun and functional gadgets with the Arduino Uno and Leonardo boards. Filled with step-by-step instructions and detailed illustrations, The TAB Book of Arduino Projects: 36 Things to Make with Shields and Proto Shields provides a cost estimate, difficulty level, and list of required components for each project.

\$30.00



Arduino Projects for Amateur Radio

by Jack Purdum, Dennis Kidder

Boost Your Ham Radio's Capabilities Using Low Cost Arduino Microcontroller Boards

Do you want to increase the functionality and value of your ham radio without spending a lot of money? This book will show you how! *Arduino Projects for Amateur Radio* is filled with step-by-step microcontroller projects you can accomplish on your own — no programming experience necessary!

Reg **\$30.00** Sale Price **\$23.95**

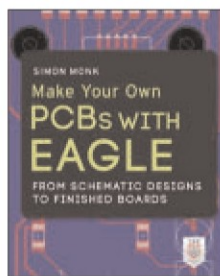


Make Your Own PCBs with EAGLE

by Eric Kleinert

Featuring detailed illustrations and step-by-step instructions, *Make Your Own PCBs with EAGLE* leads you through the process of designing a schematic and transforming it into a PCB layout. You'll then move on to fabrication via the generation of standard Gerber files for submission to a PCB manufacturing service. This practical guide offers an accessible, logical way to learn EAGLE and start producing PCBs as quickly as possible.

\$30.00



Build Your Own Transistor Radios

by Ronald Quan

A Hobbyist's Guide to High Performance and Low-Powered Radio Circuits

Create sophisticated transistor radios that are inexpensive yet highly efficient. Inside this book, it offers complete projects with detailed schematics and insights on how the radios were designed. Learn how to choose components, construct the different types of radios, and troubleshoot your work.

*Paperback, 496 pages

\$49.95

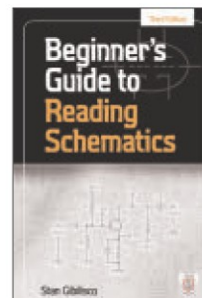


Beginner's Guide to Reading Schematics, 3E

by Stan Gibilisco

Navigate the roadmaps of simple electronic circuits and complex systems with help from an experienced engineer. With all-new art and demo circuits you can build, this hands-on, illustrated guide explains how to understand and create high-precision electronics diagrams. Find out how to identify parts and connections, decipher element ratings, and apply diagram-based information in your own projects.

\$25.00



How to Diagnose and Fix Everything Electronic

by Michael Jay Geier

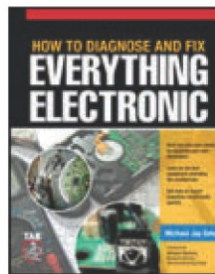
Master the Art of Electronics Repair

In this hands-on guide, a lifelong electronics repair guru shares his tested techniques and invaluable insights.

How to Diagnose and Fix Everything

Electronic shows you how to repair and extend the life of all kinds of solid-state devices, from modern digital gadgetry to cherished analog products of yesteryear.

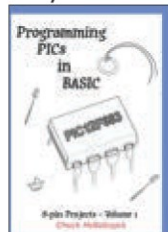
\$24.95



Programming PICs in Basic

by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you've found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller.

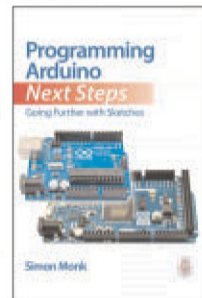


\$14.95

Programming Arduino Next Steps: Going Further with Sketches

by Simon Monk

In this practical guide, electronics guru Simon Monk takes you under the hood of Arduino and reveals professional programming secrets. Also shows you how to use interrupts, manage memory, program for the Internet, maximize serial communications, perform digital signal processing, and much more. All of the 75+ example sketches featured in the book are available for download.



\$20.00

Order online @ www.store.nutsvolts.com
Or CALL 1-800-783-4624 today!

EDUCATIONAL

Beginners Guide Book Combo.



Only \$85.95
Plus
FREE Priority Mail Shipping
US Only

An Arduino Workshop

Are you puzzled about the Arduino but finding it difficult to get all the pieces in one place?

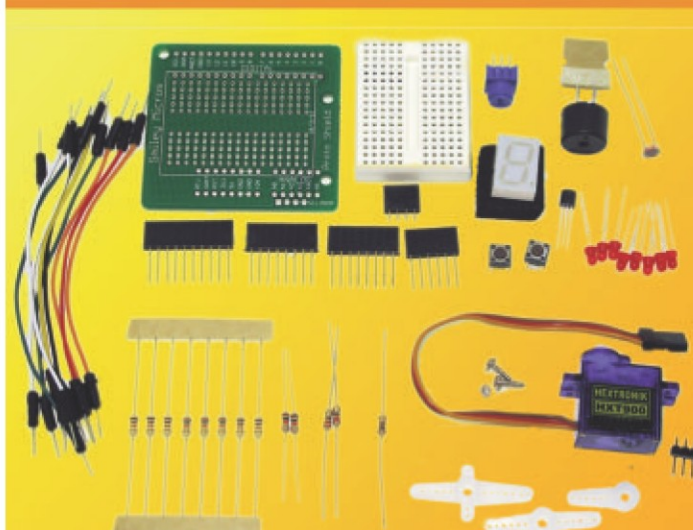


Joe Pardue
SmileyMicros.com

Book and Kit Combo \$124.95



Arduino Classroom Arduino 101 Projects Kit



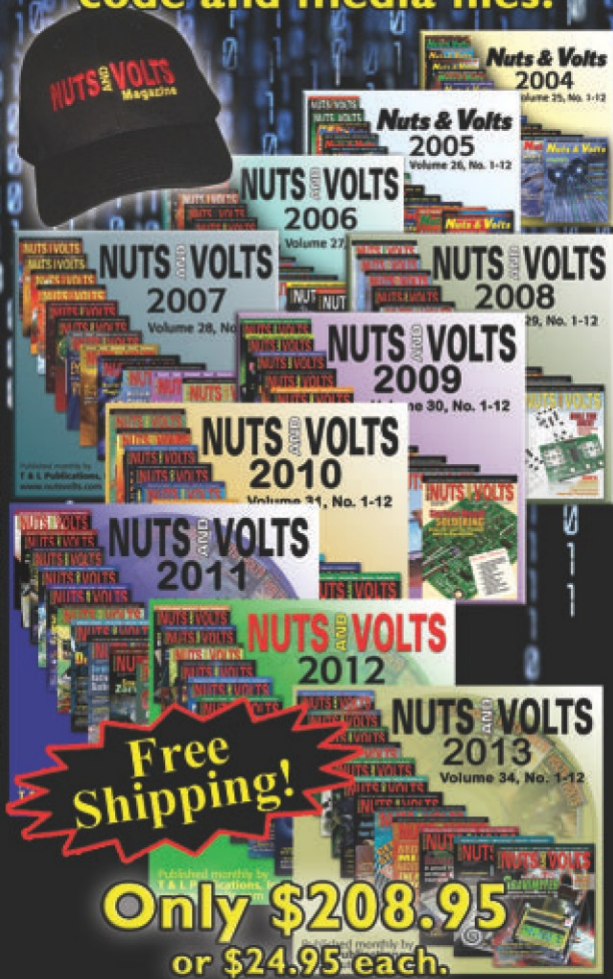
\$44.99

From Smiley's Workshop

CD-ROM SPECIAL

Nuts & Volts 10 CD-ROMs & Hat Special!

That's 120 issues.
Complete with supporting
code and media files.



Free Shipping!

Only \$208.95
or \$24.95 each.



**The Nuts & Volts
Pocket Ref**

All the info you need at your fingertips!

This great little book is a concise
all-purpose reference featuring
hundreds of tables, maps,
formulas, constants &
conversions.

Only \$12.95

AND it still fits in
your shirt pocket!

Visit <http://store.nutsvolts.com> or call (800) 783-4624

CALL 1-800-783-4624 today!
Order online @ www.nutsvolts.com

PROJECTS

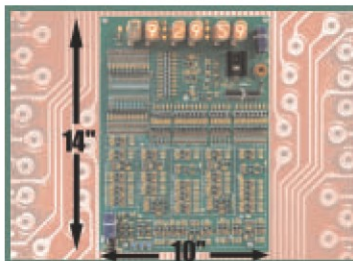
Accelerometer Kit



The Hockey Puck accelerometer is a compact digital G-force recorder. Find out if your packages are being handled with care (or not!) by your shipping company. See if your crated animals are being well treated on airplane flights. See how many Gs are being applied to the groceries in the trunk of your car. Tie it to the end of a string and see how many Gs you can generate swinging it by hand.

\$59.95

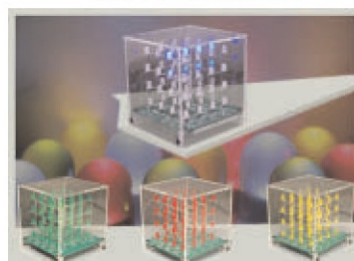
Neon Transistor Clock Kit



Add HIGH VOLTAGE to your clock!
 This is a Nixie Tube display version of the Transistor Clock. It uses only discrete components — no integrated circuits. Approximately 1,250 of them!
 This kit is a "must have" for the analog electronic hobbyist.
 For more info, see the April 2012 issue or download the 87 page manual from our webstore.

Sale Price \$245.95

3D LED Cube Kit



This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes. Colors available: Green, Red, Yellow & Blue

\$57.95

Solar Charge Controller Kit 2.0



If you charge batteries using solar panels, then you can't afford not to have them protected from over-charging. This 12 volt/12 amp charge controller is great protection for the money. It is simple to build, ideal for the novice, and no special tools are needed other than a soldering iron and a 9/64" drill!

\$27.95

Geiger Counter Kit

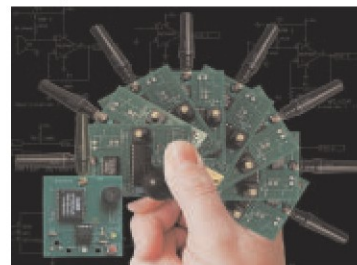


This kit is a great project for high school and university students. The unit detects and displays levels of radiation, and can detect and display dosage levels as low as one micro-roentgen/hr. The LND 712 tube in our kit is capable of measuring alpha, beta, and gamma particles.

Partial kits also available.

\$159.95

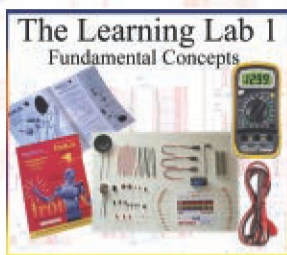
Super Detector Circuit Set



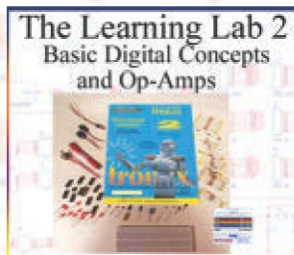
Pick a circuit!
 With one PCB you have the option of detecting wirelessly: temperature, vibration, light, sound, motion, normally open switch, normally closed switch, any varying resistor input, voltage input, mA input, and tilt, just to name a few.

\$32.95

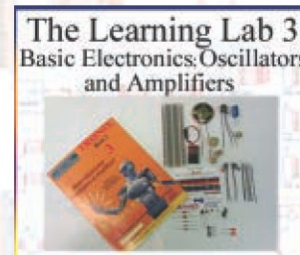
FOR BEGINNER GEEKS!



\$59.95



\$49.95



\$39.95

These labs from LF Components show simple and interesting experiments and lessons, all done on a solderless circuit board.

As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal of knowledge with each successive experiment.

For more info and lab details, please visit our webstore.

CLASSIFIEDS

SURPLUS

SURPLUS ELECTRONIC PARTS & ACCESSORIES



Over
20,000
Items
In
Stock

Belts
Cables
Connectors
Fans

Hardware
LEDs
Motors
Potentiometers

Relays
Semiconductors
Service Manuals
Speakers

Switches
Test Equipment
Tools
VCR Parts

Surplus Material Components
SMC ELECTRONICS
www.smcelectronics.com

No Minimum Order.
Credit Cards and PAYPAL Accepted.
Flat \$4.95 per order USA Shipping.

SERVICES



Sandtronic Electronics Services LLC

"design of custom electronics instrumentation"

Circuit design & drawings, PCB layout, fabrication & population
Prototype development for research, industry and inventors.
39 years experience in electronics development for research &
flight applications, laser systems & laboratory research

FREE no obligation consultations

www.sandtronic.com

HARDWARE WANTED

DEC EQUIPMENT WANTED!!!

Digital Equipment Corp.
and compatibles.
Buy - Sell - Trade

CALL KEYWAYS 937-847-2300
or email buyer@keyways.com

KITS/PLANS



QKITS LTD
sales@qkits.com

1 888 GO 4 KITS

Arduino • Raspberry Pi
Power Supplies
MG Chemicals
RFID

\$8.50
flat rate
shipping

Visit us at:

www.qkits.com

LIGHTING

ROBOTICS

MaxBotix
High Performance Ultrasonic Rangefinders

Check us out Today!

4-20HR-MaxSonar®-WR™

- 4-20mA output
- High noise tolerance
- IP67 rated
- 1.6 mm resolution
- Multi-Sensor operation
- Calibrated beam pattern
- 8Hz head rate
- Ideal for industrial use



UCXL-MaxSonar®-WR™

- Great for design engineers
- Multiple mounting options
- Light weight industrial sensor
- Incredible noise immunity
- Smallest IP67 sensor in size
- 1mm resolution
- Automatic calibration
- Ideal for outdoor UAV's

Phone: 218-454-0786

Email: sales@maxbotix.com

www.maxbotix.com



Automatic
Stair Lighting
www.ReactiveLighting.com

AUDIO/VIDEO PARTS

ICEpower MODULES

50W x 2 (50ASX2) or
170W x 1 (50ASX2BTL)
Just add AC & speaker(s)
\$99 USA \$119 Worldwide
www.classicaudioparts.com

PARTS
EXPRESS
YOUR ELECTRONICS CONNECTION™

OVER 18,000
ELECTRONIC PARTS
IN STOCK



Speakers



Components



Project
Accessories



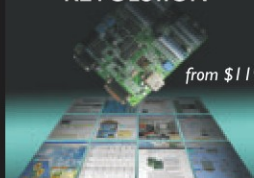
Tools and
Tech Aids

Call or visit us online today to receive
your **FREE** copy of our 2015 catalog!

parts-express.com/nuts
1-800-338-0531

CONTROLLERS

Join The
INTERNET of THINGS
REVOLUTION



from \$119

TRI SUPER PLCs

Powerful & Easy Loader
+BASIC Programming
Ethernet Integrated
MODBUS TCP/IP
DIOs & AIOs integrated

tel : 1 877 TRI-PLCS
web : www.triplc.com/nx.htm

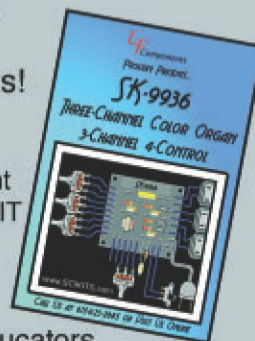
TRI TRIANGLE
RESEARCH
INTERNATIONAL

FREE SAMPLE KIT*



Check out our full line of kits!

We also carry the full line of
TRON.IX Labs and replacement
parts for TRON.IX & MR.CIRCUIT
LAB SETS
www.LFComponents.com



*Free Kit offer for ALL Teachers/Educators

Visit LFComponents.com/FREE or call to receive your free kit today

(Free Kit will be either a SK9914 or SK9999)

We are so confident you'll love what we have to offer that we'll send
you the first kit free!

ASK ABOUT THE ALL NEW FOR
2015 TRONIX LAB SYSTEMS!
Available February 2015
NOW TAKING PRE-ORDERS

LF Components
615-625-2885

>>> QUESTIONS

IC Identification

Figure 1 is the front-end of a simple frequency counter, 1 Hz to 1 MHz. I can't seem to find IC1 4583, (I'm guessing this is a CD4583). Pin 4 of IC1 feeds the input of a CD4026. Q1 is a 2N930 and IC5A is a 555.

I found two conflicting datasheets on this. One says it's a microcontroller (which I doubt); the other says it's a flip-flop. I've looked at Mouser and Digi-Key and can't find this IC, even at some surplus places I've looked!

Can you help?

#1151

Ted Mieske
CA, USA

PointSource/Qioptiq Pin-Out

I purchased a model P1600-Gemini PC-4 mfg. by PointSource which is now called Qioptiq. It is a three laser diode device, 660-680 nm. It has three fiber steel shielded optic cables. It has two input ports, 26-pin and nine-pin. I think the 26-pin port is for control and monitoring. The nine-pin I think is for power in. I cannot find the pin-out specifications to be able to power up. Emailed Qioptiq ... no reply.

This device will be used in a quantum entanglement project if I can get the pin-outs. Any information would be extremely helpful.

#1152

Michael Duc
Vancouver, WA

HD44780-based LED Display: Scrolling Text False Start

I have been experimenting with scrolling text on some HD44780-based LED displays of various sizes. The commands I talk about below are those for that device, not any particular PIC code.

I first clear the display and home the cursor sending a *01* command to the display. Then, I change to One Line mode with a *48* command. A long text message is then sent to the display using the data mode in a loop to send all the characters. This message is 72 bytes long and should be going into LCD RAM locations 128 through 199.

At this point, I have tried a number of commands to reliably reset the active window to the start of my message, but none of them seem to work for all the different displays I have tried this on.

On some of the displays, nothing works. Then, the text is scrolled using the Pan Right command: 24 in a loop

R1 = 8.2M

R2, R9 = 100K

R3 = 470K

R4 = 470 ohm

R5, R6, R7 = 10K

R8 = 3.3M

C1, C2 = 1µF 63V Mylar NPO

C3 = 47µF 16V

C4 = .1µF 63V

C5 = 2.2µF 16V C6 = 10µF 16V

C7 = .01µF 63V Mylar

C8, C10 = .001µF 63V Mylar

C9 = 1µF 16V

TR1 = 1M trimmer 10T

TR2 = 1K trimmer 10T

Q1 = 2N930 or 2N2222A

IC1 = CD4583

IC2, IC3, IC4 = CD4026

IC5 = LM555

IC6 = CD4007

IC7 = 7805

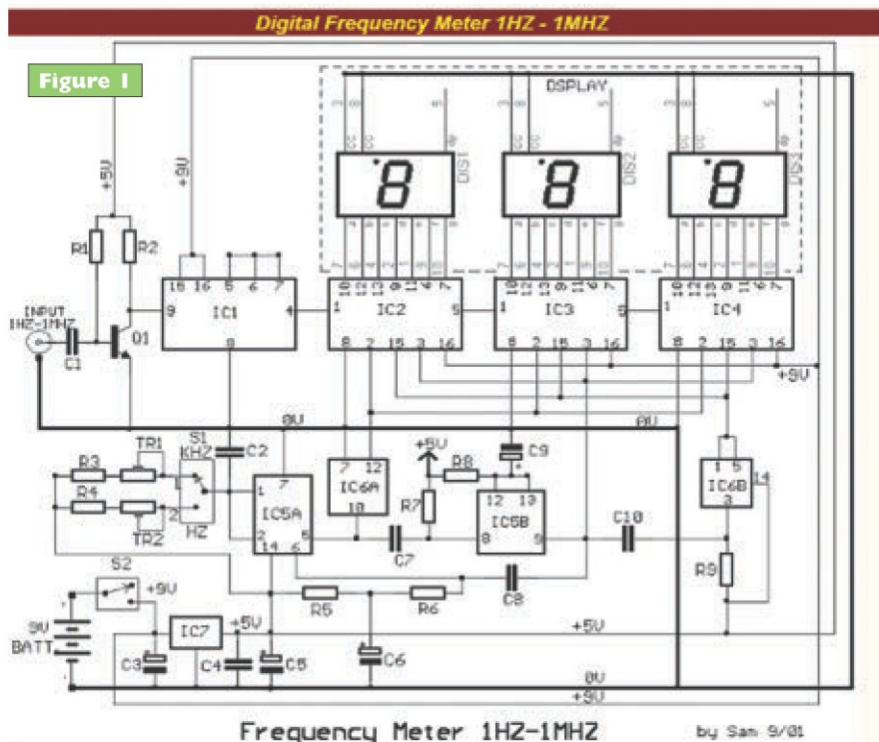
DS1, DS3 = Display 7 Seg. Comm. Cath.

S1 = ON-OFF mini switch

S2 = 1X2 mini switch

9V Battery

Frequency Meter



All questions AND answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. All submissions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted

by readers and **NO GUARANTEES WHATSOEVER** are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgment!

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

that repeats it 136 times to display the complete message two times. The scrolling does work, but sometimes it starts at the beginning of the message and other times it starts at the 41st character. This action seems to be random.

As I stated, I have tried inserting several different commands just before the scrolling routine but none of them have worked. These commands include repositioning the cursor to position 128, the start of the first line, homing the cursor (2), and adding a cursor right and a cursor left command after that. None of these measures seems to have any effect.

Does anybody know of a reliable way to have this scrolling message always start at its beginning? If possible, I need a solution that works on all HD44780-based displays. I know this is a tall task and may need to use different approaches for different displays.

#1153

Paul Alciatore
Beaumont, TX

>>> ANSWERS

[#12142 - December 2014]
Unregulated Power Supply

I picked up a 9V switching power supply wart from one of the online suppliers that advertise in Nuts & Volts. Unfortunately, it melted two Arduinos before I realized what was happening. Apparently, the no-load voltage on a switching supply can be up to twice

the rated voltage — in my case, 18V. Should I add a load resistor to the internal circuit of the wart so there's always a load? Any other suggestions?

All you need is an appropriate three-terminal voltage regulator to protect your Arduino boards. I don't know what voltage your board needs, so I suggest integrating an LM317T programmable regulator between the board's power jack and the board. The regulator will take the wall wart's high voltage and reduce it to what the board requires. To program the regulator, connect it between the wall wart and your board's DC input using the **Figure 2** diagram as a guide.

Wiring isn't too critical (point-to-point), but keep lead lengths as short as practical. Be sure to include the two capacitors as they'll help "keep things stable" and prevent unwanted high frequency oscillations. **NOTE:** The voltage rating of C2 should be above 24V as it's typically an electrolytic. Variable resistor R2 will let you dial in the output to what your board requires. **BE AWARE:** Keep

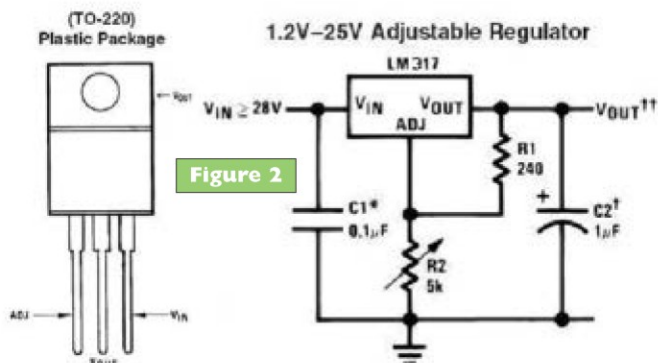
the maximum current draw BELOW one amp; use a heatsink on the regulator (a TO-220 type clamp-on will be sufficient) and keep the regulator's input voltage below 35 VDC!

If you don't want to bother with a variable resistor for R2, visit www.reuk.co.uk/LM317-Voltage-Calculator.htm as it has a nice table of standard resistor values to use for R1 and R2 for programming various output voltages (and other good info on the regulator).

Ken Simmons
Auburn, WA

[#12143 - December 2014]
Photomicrographs

I need photomicrographs of some 3D printed surfaces for a science fair poster. Is there a commercial electron microscope imaging service that you



What is a Plasma Arc Speaker?



The Plasma Arc Speaker has a 3.5 MM audio input jack for use with MP3 players, computers, phones, etc. Uses a 12 VDC power supply, included.

PAS-01K \$89.95

www.imagesco.com/kits/plasma-speaker.html

Plasma is a hot ionized gas that is electrically conductive. The plasma arc speaker creates a plasma using a high voltage electrical discharge. When an audio signal is passed through the plasma it causes it to vibrate in sync with the audio signal. The plasma in turn vibrates the air around it to create sound.

Fascinating to watch!

IMAGES
SCIENTIFIC INSTRUMENTS

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

can recommend? I tried my local community college, but they weren't willing to help.

This is in response to the Fabian Hoffmann question about electron micrographs. We perform limited "community service imaging" of the type Fabian describes for school projects. We would need to know the scope of the project (usually just a few micrographs). Do let us know of your interest; www.marshall.edu/mbic/.

Mike Norton
via email

**[#10141 - October 2014]
555 for PWM**

Does anyone have a simple circuit/schematic to use a 555 timer to dim a 120 VAC incandescent bulb? In the little bit of research I've done, it seems like using the 555 for PWM should do the trick, but I don't understand "zero crossing" and why it's important. Pointers welcome!

I have a 555 PWM circuit diagram on one of my YouTube videos at <http://youtu.be/KrcqpGZQd9U?list=PLUsDZdokq6wxDP7L3sMoYFTA5WDoAtBu4>. The diagram is at about 2:18 in the video. I hope it helps.

Freddy Cordero
via email

**[#12141 - December 2014]
Nixie Tube Specs**

I have six Fritron DG10A Nixie tubes. I need to find specs and pin-outs for these tubes to see if they are okay. Any info is appreciated.

The DG10A is not a Nixie tube but is a seven-segment vacuum fluorescent tube. This vacuum fluorescent readout operates much in the fashion of a triode vacuum tube. A heated filament/cathode emits electrons that are attracted to any of the seven anodes which are biased with a positive voltage. Each anode is coated with a material that fluoresces when the electrons strike the anode.

Changing biasing on the control grid varies the intensity of the electron flow, and therefore the brightness of the display. The filament requires about 50 milliamperes, which operates just below its incandescent level. About 0.8 VDC will provide the required current. Apply about 50 VDC to each anode (segment) that you want to turn on. For test purposes, the common control grid may also be biased at 50 VDC to provide maximum brightness. The pin-out for the DG10A can be found at www.radio-museum.org/tubes/tube_dg10A.html.

Fred Franke
Kings Park, NY

**[#11143 - November 2014]
QuickBASIC Program**

I thought that maybe a proficient N&V reader could help me with a sorting program by correctly coding in Microsoft QuickBASIC.

What I'd like to accomplish is by electing choice (1), sort in ascending order just the six-numbered string, paired with the correct and corresponding "date" and "odd/even" strings. Therefore, the six-numbered string is sorted in ascending order while the corresponding date and odd/even string must stay with that six-numbered string (the dates and odd/even strings must also be swapped).

Also, I'd like to print to the screen – bit by bit/segment by segment – the entire generated output by depressing any keyboard button to advance to the next screen segment until I reach the end of the generated sortition. The way I've coded it, it just goes automatically to the last part of the screen output, without being able to ever visualize either the start nor the middle of the generated screen output.

As for choice (2), I'd like to only sort the string dates – in ascending order – paired with the correct six-numbered strings and corresponding odd/even strings. Lastly, I'd like to print to screen (just as for choice 1 – bit by bit) by depressing some button in order to advance to the next screen segment,

until the entire generated sortition would have appeared on the screen.

#1 I don't have a quick answer for you, but the version of QuickBASIC you are using is quite out of date. Of the BASICs running on micros these days, I don't think any still use line numbers, REM statements, and the like. The language is more structured with subroutine and function names. I too cut my teeth on HP educational BASIC from that era, but really don't miss RENUM as a edit feature. Now, you compose programs in an editor and download them to a target. Our BASIC which targets various ARM processors can also be run on a PC in emulation mode, so you can start out writing programs there. Go to www.coridium.us/files/setupBASIC.exe.

Bruce Eisenhard
Tahoe Vista, CA

#2 Don, I'm not a QuickBASIC user, but what I see is you need matching SWAP statements (just like you say) in 290 & 347 when true, SWAP N\$, SWAP B\$, SWAP C\$, then your arrays should stay in sync.

I'm not sure I follow your screen print requirement. However, the INPUT statement should allow you to both print an output and then wait for the Enter key to be pressed before continuing. Place the statement at the point you want to interrupt. One more thing: The loop 330-339 will print the same value for D\$ over and over; you want to move 311 to 331.

Doug Arndt
Chicago, IL

#3 Visit the Tech Forum section at www.nutsvolts.com for additional answers to this and other questions, including some well-documented code from **Raymond Perry of Jacksonville, FL**. While you're there, be sure to submit your tech questions for our readers to test their knowledge and skills!

BGA device can be soldered to the male adapter module. Both modules are constructed with high temperature FR-4 substrate, assuring a match with target PCBs and preventing failures that occur with CTE mismatch. The male adapter module with the BGA device is plugged into the SF-BGA724A-B-42F on the target board so the device and motherboard are interconnected and the system is ready to go. The modules come with both lead and lead-free solder ball type options. The BGA socket adapter pair requires half the force of conventional connectors at 20 pounds for the 724-pin interface for ease of operation.

The electrical path of the BGA socket adapter is a high priority performance issue; the physical length from

the top connection point on the male pin module to the solder ball on the female socket is 4.5 mm. This is one of the shortest connection lengths by far for machined pin sockets, therefore providing better transmission of high frequency signals. The BGA socket adapter line is available in many different pin counts with 1 mm, 0.8 mm, and 0.5 mm pitch. Pricing for the SF-BGA724A-B-42F (quantity 1) is \$229, with reduced pricing available depending on quantities required.

For more information, contact:
Ironwood Electronics
 Web: www.ironwoodelectronics.com

■ LOOK FOR ■ SEARCH FOR ■ FIND

Find your favorite advertisers here!

ADvertiser INDEX

AMATEUR RADIO AND TV

Alpha Node Hub22
 EZ Hang25
 National RF25
 Ramsey Electronics82-83

ANTENNA

Alpha Node Hub22

BATTERIES/CHARGERS

Cunard Associates25

BUYING ELECTRONIC SURPLUS

Earth Computer Technologies6
 Weirdstuff Warehouse25

CCD CAMERAS/VIDEO

Ramsey Electronics82-83

CIRCUIT BOARDS

AP Circuits10
 Cunard Associates25
 Dimension Engineering65
 ExpressPCB17
 Front Panel Express LLC42
 Little Bits
 ElectronicsBack Cover
 PCB Pool47
 Saelig Co. Inc.11

COMPONENTS

Anaren67
 Electronix Express51
 LF Components77

COMPUTER

Hardware

Earth Computer Technologies6
 Weirdstuff Warehouse25

Microcontrollers / I/O Boards

Images Co.79
 M.E. Labs64
 MikroElektronika3

Parallax, Inc.42
 Technologic Systems18

DESIGN/ENGINEERING/REPAIR SERVICES

Cleveland Institute of Electronics.22
 ExpressPCB17
 Front Panel Express LLC42
 Images Co.79
 National RF25
 PCB Pool47

DIGITAL OSCILLOSCOPES

Rigol Technologies6

EDUCATION

Boxed Kit Amps25
 Cleveland Institute of Electronics.22
 Command Productions9
 LF Components77
 Little Bits
 ElectronicsBack Cover
 Lumidax11
 NKC Electronics25
 Poscope65

EMBEDDED TOOLS

NetBurner2

ENCLOSURES

Front Panel Express LLC42

HI-FI AUDIO

Boxed Kit Amps25

KITS & PLANS

Boxed Kit Amps25
 Earth Computer Technologies6
 LF Components77
 NetBurner2
 NKC Electronics25
 Ramsey Electronics82-83

MISC./SURPLUS

All Electronics Corp.7
 Front Panel Express LLC42
 Weirdstuff Warehouse25

PROGRAMMERS

Little Bits
 ElectronicsBack Cover
 M.E. Labs64
 MikroElektronika3

RF TRANSMITTERS/RECEIVERS

Anaren67
 National RF25

ROBOTICS

CH Robotics23
 Cleveland Institute of Electronics.22
 Lemos International Co.25
 Robot Power25

SATELLITE

Lemos International Co.25

SENSORS

CH Robotics23

TEST EQUIPMENT

Dimension Engineering65
 Images Co.79
 Lumidax11
 NKC Electronics25
 Pico Technology5
 Poscope65
 Rigol Technologies6
 Saelig Co. Inc.11

TOOLS

EZ Hang25
 MikroElektronika3
 NetBurner2
 PanaVise10
 Poscope65

VIDEO/DISPLAY MODULES

4D Systems39

WAVEFORM GENERATORS

Rigol Technologies6

WIRELESS PRODUCTS

Anaren67

4D Systems39
 All Electronics Corp.7
 Alpha Node Hub22
 Anaren67
 AP Circuits10
 Boxed Kit Amps25
 CH Robotics23
 Cleveland Institute of
 Electronics22
 Command Productions9
 Cunard Associates25
 Dimension Engineering.....65
 Earth Computer Technologies .6
 Electronix Express51
 ExpressPCB17
 EZ Hang25
 Front Panel Express LLC ..42
 Images Co.79
 Little Bits
 ElectronicsBack Cover
 Lemos International Co.25
 LF Components77
 Lumidax.....11
 M.E. Labs.....64
 MikroElektronika3
 National RF25
 NetBurner2
 NKC Electronics25
 PanaVise10
 Parallax, Inc.42
 PCB Pool47
 Pico Technology5
 Poscope65
 Ramsey Electronics82-83
 Rigol Technologies6
 Robot Power25
 Saelig Co. Inc.11
 Technologic Systems18
 Weirdstuff Warehouse25

HAPPY NEW YEAR!

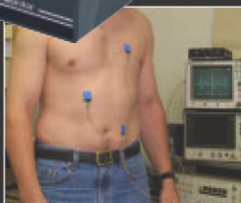
Electrocardiogram ECG Heart Monitor

- ✓ Visible and audible display of your heart rhythm!
- ✓ Bright LED "Beat" indicator for easy viewing!
- ✓ Re-usable hospital grade sensors included!
- ✓ Monitor output for professional scope display
- ✓ Simple and safe 9V battery operation



When we think of a New Year, we often start with our proverbial list of New Year's resolutions. Often starting with weight loss goals as a result of all the December holiday parties, (including the Buffalo Wings of course!) they typically also include a new attitude for a healthier life style. While we are frequently reminded that February is national Heart Smart month, we think the first of the Year is no better time to start! Every month should be Heart Smart month. Heart Smart is a way of life, and certainly shouldn't be limited to one month a year. We kept that in mind when we designed the ECG1!

Not only will building an actual ECG be a thrill, but you'll get hands-on knowledge of the relationship between electrical activity and the human body. Each time the human heart beats, the heart muscle causes small electrical changes across your skin. By monitoring and amplifying these changes, the ECG1C detects the heartbeat and allows you to accurately display it, and hear it, giving you a window into the inner workings of the human heart and body!



One of our engineers/guinea pigs, checking his heart!

Use the ECG1C to astound your physician with your knowledge of ECG/EKG systems. Enjoy learning about the inner workings of the heart while, at the same time, covering the stage-by-stage electronic circuit theory used in the kit to monitor it. The three probe wire pick-ups allow for easy application and experimentation without the cumbersome harness normally associated with ECG monitors.

The documentation with the ECG1C covers everything from the circuit description of the kit to the circuit description of the heart! Multiple "beat" indicators include a bright front panel LED that flashes with the actions of the heart along with an adjustable level audio speaker output that supports both mono and stereo hook-ups. In addition, a monitor output is provided to connect to any standard oscilloscope to view the traditional style ECG/EKG waveforms just like you see in a real ER or on one of the medical TV shows!



Look what I found!

On a personal note... See the display to the left? That's me! In between writing this monthly ad copy, catalog copy, and plethora of other tasks here, I noticed some skipped beats in my pulse! An immediate cardiac check found I had something called Trigeminy, or PVCs that occur at intervals of 2 normal beats to one PVC! And I saw it with our ECG1 kit!

The fully adjustable gain control on the front panel allows the user to custom tune the differential signal picked up by the probes giving you a perfect reading and display every time! 10 hospital grade re-usable probe patches are included together with the matching custom case set shown. Additional patches are available in 10-packs. Operates on a standard 9VDC battery (not included) for safe and simple operation.

Note, while the ECG1C professionally monitors and displays your heart rhythms and functions, it is intended for hobbyist usage only. If you experience any cardiac symptoms, seek proper medical help immediately!

ECG1C	Electrocardiogram Heart Monitor Kit With Case & Patches	\$44.95
ECG1WT	Electrocardiogram Heart Monitor, Factory Assembled & Tested	\$89.95
ECGP10	Electrocardiogram Re-Usable Probe Patches, 10-Pack	\$4.95

Digital Voice Changer

This voice changer kit is a riot! Just like the expensive units you hear the DJ's use, it changes your voice with a multitude of effects! You can sound just like a robot, you can even add vibrato to your voice! 1.5W speaker output plus a line level output! Runs on a standard 9V battery.

MK171 Voice Changer Kit \$9.95

Precision PC Plane Antennas

Our LPY series PC antennas continue to be the favorite for virtually all RF and wireless applications. From microwave links, wireless mics, to RFID, we've got you covered. Check our site for details!

LPYSeries Precision PC Plane Antennas from \$29.95

Laser Trip Senser Alarm

True laser protects over 500 yards! At last within the reach of the hobbyist, this neat kit uses a standard laser pointer (included) to provide both audible and visual alert of a broken path. 5A relay makes it simple to interface! Breakaway board to separate sections.

LTS1 Laser Trip Sensor Alarm Kit \$29.95

Steam Engine & Whistle

Simulates the sound of a vintage steam engine locomotive and whistle! Also provides variable "engine speed" as well as volume, and at the touch of a button the steam whistle blows! Includes speaker. Runs on a standard 9V battery.

MK134 Steam Engine & Whistle Kit \$11.95

Audio Recorder & Player

Record and playback up to 8 minutes of messages from this little board! Built-in condenser mic plus line input, line & speaker outputs. Adjustable sample rate for recording quality. 4-switch operation that can be remote controlled! Runs on 9-12VDC at 500mA.

K8094 Audio Recorder/Player Kit \$32.95

Electronic Watch Dog

A barking dog on a PC board! And you don't have to feed it! Generates 2 different selectable barking dog sounds. Plus a built-in mic senses noise and can be set to bark when it hears it! Adjustable sensitivity! Unlike my dog, eats 2-8VAC or 9-12VDC, it's not fussy!

K2655 Electronic Watch Dog Kit \$39.95

The Learning Center!



Beginners To Advanced... It's Fun!

- ✓ Learn, build, and enjoy!
- ✓ 130, 200, 300, & 500 in one electronic labs!
- ✓ Practical through hole and SMT soldering labs!
- ✓ Integrated circuit AM/FM radio lab!
- ✓ AM, AM/FM, and SWL radio labs!
- ✓ Radio Controlled (RC) car!
- ✓ Beginner's non-soldering kits!

For over 4 decades we've become famous for making electronics fun, while at the same time making it a great learning experience. As technology has changed over these years, we have continued that goal!

PL130A Gives you 130 different electronic projects together with a comprehensive learning manual describing the theory behind all the projects.

PL200 Includes 200 very creative fun projects and includes a neat interactive front panel with 2 controls, speaker, LED display and a meter.

PL300 Jump up to 300 separate projects that start walking you through the learning phase of digital electronics.

PL500 The ultimate electronics lab that includes 500 separate projects that cover it all, from the basics all the way to digital programming.

SP1A Whether young or old, there's always a need to hone your soldering skills. Either learn from scratch or consider it a refresher, and end up with a neat little project when you're done!

SM200K Move up to Surface Mount Technology (SMT) soldering, and learn exactly how to solder those tiny little components to a board!

AMFM108K We not only take you through AM and FM radio theory but we guide you through IC's. When you're done you've built yourself an IC based AM/FM radio that works great!

AM2 Learn the complete theory of AM broadcast radio and end up with a highly sensitive AM radio receiver!

AMFM7 Step up to AM/FM with this multi-band lab and learn the basics of both bands. The FM tuner is factory assembled and aligned!

SR3 Enter the world of SWL with this short-wave radio learning lab covering 6-8MHz and 12-18MHz!

AK870 One of the most exciting electronic learning kits that the kids will love! Build a complete RC speedster from the ground up! 7 remote functions!

PL130A	130-In-One Lab Kit	\$39.95
PL200	200-In-One Lab Kit	\$84.95
PL300	300-In-One Lab Kit	\$109.95
PL500	500-In-One Lab Kit	\$249.95
SP1A	Through Hole Soldering Lab	\$9.95
SM200K	SMT Practical Soldering Lab	\$22.95
AMFM108K	AM/FM IC Lab Kit & Course	\$36.95
AM2	AM Radio Learning Lab	\$11.95
AMFM7	AM/FM Radio Learning Lab	\$12.95
SR3	Short Wave Radio Learning Lab	\$16.95
AK870	RC Speedster Car Kit	\$29.95



Stereo Audio Platform Gain Controller

- ✓ Stereo audio processing while preserving audio dynamics!
- ✓ True stereo control keeps virtual sonic source location intact!
- ✓ Auto-bypass restores original levels when power is turned off!
- ✓ Built-in bar graph indication of signal level with display mute!

The SGC1 is one of our latest kits, and provides a great solution to the age-old problem: how can we easily correct inconsistent audio levels without negatively affecting the dynamics of the audio signal? The SGC1 circuit implements a principle known as the "Platform Gain Principle," which was originally developed by CBS Labs to allow transmitted audio levels to be automatically adjusted to keep them within a desired range.

Think of it like an audio engineer, constantly adjusting the output level in order to limit highs that would be too loud while boosting lower levels so that they can still be heard. You may think "oh, this is just another limiter/compressor!" Not so! Here's the real trick: keeping the full dynamic range ratio of the output signal the same as the original input - something the typical limiter/compressor can only dream of doing! The SGC1 can be placed in just about any standard analog stereo line level audio circuit to keep the audio level within the desired range. It's also the perfect addition to any of our hobby kit transmitters, allowing you to match levels between different audio sources while keeping lows audible and preventing the highs from overdriving.

In addition to its useful basic function and great audio performance, the SGC1 also boasts a front panel LED meter to give an indication of the relative level of the input signal, as well as a front level control that allows you to adjust the controller to the min/max center point of your desired level range. And yes, it is a **Stereo Gain Controller!** Meaning that the levels of both the left and right channels are monitored and adjusted equally, thereby maintaining the relative virtual position of things like instruments, singers and speakers! The entire unit is housed in a slim attractive black textured aluminum case that is sure to complement your studio or home theatre. If you're looking for perfect audio levels, hire a broadcast audio engineer, but if that doesn't fit your budget, the SGC1 is the next best thing! Includes 15VDC world-wide power adapter.

SGC1 Stereo Audio Platform Gain Controller Kit

\$179.95

RF Preamplifier

The famous RF preamp that's been written up in the radio & electronics magazines! This super broadband preamp covers 100 KHz to 1000 MHz! Unconditionally stable gain is greater than 16dB while noise is less than 4dB! 50-75 ohm input. Runs on 12-15 VDC.

SA7 RF Preamp Kit

\$16.95

Mad Blaster Warble Alarm

If you need to simply get attention, the "Mad Blaster" is the answer, producing a LOUD ear shattering raucous racket! Super for car and home alarms as well. Drives any speaker. Runs on 9-12VDC.

MB1 Mad Blaster Warble Alarm Kit

\$9.95

Water Sensor Alarm

This little \$7 kit can really "bail you out"! Simply mount the alarm where you want to detect water level problems (sump pump)! When the water touches the contacts the alarm goes off! Sensor can even be remotely located. Runs on a standard 9V battery.

MK108 Water Sensor Alarm Kit

\$6.95

Air Blasting Ion Generator

Generates negative ions along with a hefty blast of fresh air, all without any noise! The steady state DC voltage generates 7.5kV DC negative at 400uA, and that's LOTS of ions! Includes 7 wind tubes for max air! Runs on 12-15VDC.

IG7 Ion Generator Kit

\$64.95

Tri-Field Meter Kit

"See" electrical, magnetic, and RF fields as a graphical LED display on the front panel! Use it to detect these fields in your house, find RF sources, you name it. Featured on CBS's Ghost Whisperer to detect the presence of ghosts! Req's 4 AAA batteries.

TFM3C Tri-Field Meter Kit

\$74.95

Electret Condenser Mic

This extremely sensitive 3/8" mic has a built-in FET preamplifier! It's a great replacement mic, or a perfect answer to add a mic to your project. Powered by 3-15VDC, and we even include coupling cap and a current limiting resistor! Extremely popular!

MC1 Mini Electret Condenser Mic Kit

\$3.95

Touch Switch

Touch on, touch off, or momentary touch hold, it's your choice with this little kit! Uses CMOS technology. Actually includes TWO totally separate touch circuits on the board! Drives any low voltage load up to 100mA. Runs on 6-12 VDC.

TS1 Touch Switch Kit

\$9.95

Laser Light Show

Just like the big concerts, you can impress your friends with your own laser light show! Audio input modulates the laser display to your favorite music! Adjustable pattern & speed. Runs on 6-12VDC.

LLS1 Laser Light Show Kit

\$49.95

20 Watt Mini Audio Amp

Delivers a super clean 20W output from one SMT package! Ultra efficient class D design produces no heat. PCB can be snapped into a small circle for special applications. Runs on 18VDC for rated output, or down to 10VDC for reduced output.

UAM2 20W Subminiature Amp Kit

\$52.95

Tickle-Stick Shocker

The kit has a pulsing 80 volt tickle output and a mischievous blinking LED. And who can resist a blinking light and an unlabeled switch! Great fun for your desk, "Hey, I told you not to touch!" Runs on 3-6 VDC.

TS4 Tickle Stick Kit

\$9.95

Passive Aircraft Monitor

The hit of the decade! Our patented receiver hears the entire aircraft band without any tuning! Passive design has no LO, therefore can be used on board aircraft! Perfect for airshows, hears the active traffic as it happens! Available kit or factory assembled.

ABM1 Passive Aircraft Receiver Kit

\$89.95

12VDC Regulated Supply

Go green with our new 12VDC 1A regulated supply. Worldwide input 100-240VAC with a Level-V efficiency! It gets even better, includes DUAL ferrite cores for RF and EMI suppression. All this at a 10 buck old wallwart price!

AC121 12VDC 1A Regulated Supply

\$9.95

Synthesized FM Stereo Transmitter Kit

The FM25 has been the hobbyist's standard for FM synthesized stereo transmitters for more than two decades! Just plug in the stereo left/right audio from your MP3 player, CD player, or computer, and broadcast it on any frequency in the standard FM broadcast band.

The FM25B brings stereo line level audio into the unit through a 3.5mm stereo phone jack. A loop out is provided at the input for monitoring. The unit features a PIC microprocessor for easy frequency programming through board mounted DIP switches. The transmit frequency is Phase Locked Loop (PLL) controlled for unparalleled stability making frequency drift a thing of the past, extremely critical for digital tuners. The RF output level is adjustable from 5uW to 25 mW via a potentiometer. Use the built-in whip antenna or use an external antenna with the standard "F" external antenna connector on the rear panel. Includes power supply & stereo cable. (Note: The FM25B is a do-it-yourself learning kit that you assemble. Please remember that the end user is responsible for complying with all FCC rules & regulations within the US, or any regulations of their respective governing body in regards to the application and use of the FM25B.)

FM25B Synth FM Stereo Xmtr Kit

\$139.95

3.4dB Gain FM Omni Antenna

Our 5/8 wave omni antenna has been the standard for LPFM installations worldwide. Provides 3.4dB gain while keeping the signal radiation low to the horizon for maximum range. Field tuneable over the entire FM range for a perfect match. SO239 connector.

FMA200E Omnidirectional FM Antenna

\$139.95

LED Red Blinky

Alternately flashes two jumbo red LEDs. You've seen this used on name badges, hats, buttons, and attention getters for decades! Has been the most popular 1st learning kit for kids young and old! Runs on 3-9VDC.

BL1 LED Red Blinky Kit

\$7.95

HV Plasma Generator

Generate 2" sparks to a handheld screwdriver! Light fluorescent tubes without wires! This plasma generator creates up to 25kV at 20kHz from a solid state circuit! Build plasma bulbs from regular bulbs and more! Runs on 16VAC or 5-24VDC.

PG13 HV Plasma Generator Kit

\$64.95

Speedy Speed Radar Gun

Our famous Speedy radar gun teaches you doppler effect the fun way! Digital readout displays in MPH, KPH, or FPS. You supply two coffee cans! Runs on 12VDC or our AC121 supply.

SG7 Speed Radar Gun Kit

\$74.95

Broadband RF Preamp

Need to "perk-up" your counter or other equipment to read weak signals? This preamp has low noise and yet provides 25dB gain from 1MHz to well over 1GHz. Output can reach 100mW! Runs on 12 volts AC or DC or the included 110VAC PS. Asmb.

PR2 Broadband RF Preamp

\$69.95

12VDC Worldwide Supply

It gets even better than our AC121 to the left! Now, take the regulated Level-V green supply, bump the current up to 1.25A, and include multiple blades for global country compatibility! Dual ferrite cores!

PS29 12VDC 1.25A Worldwide Supply

\$19.95

GET THE INUTS AND VOLTS DISCOUNT!

Mention or enter the coupon code **NVRMZ142** and receive **10% off your order!**

800-446-2295
www.ramseykits.com

Prices, availability, and specifications are subject to change. Not responsible for typos, mistakes, stupid, or New Year's morning headaches! Robin gets cold this year for making me get this ad copy done before the Turkey! Visit www.ramseykits.com for the latest pricing, specials, terms and conditions. Copyright 2014-2015 Ramsey Electronics™... so there!

RAMSEY ELECTRONICS®

590 Fishers Station Drive
Victor, NY 14564
(800) 446-2295
(585) 924-4560



littleBits[™]
ELECTRONICS

the bitLab[™]

USER-GENERATED HARDWARE IS HERE.

Get started at

littleBits.cc/bitLab

littleBits makes an expanding library of modular electronics that snap together with magnets. It's the easiest and most extensive way to learn and prototype with electronics. Create complex circuits in seconds: no soldering, wiring, or programming required (unless you want to).

The bitLab is the new marketplace for user-generated hardware. You can submit your own hardware designs for modules you want to see added to the littleBits library. If your design gets enough community support, we'll manufacture it, market it, sell it and give you a percentage of the sales!